Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design

http://yaoquantum.org/

Creators of Yao

Xiu-Zhe Luo, U Waterloo & PI

Jin-Guo Liu, IOP CAS

What is the killer app of a <u>near-term</u> quantum computer?

↓

In about next 3 years

Small: $O(10)$-$O(10^3)$ qubits

Shallow: $O(10^2)$-$O(10^4)$ gates

Noisy: no error correction

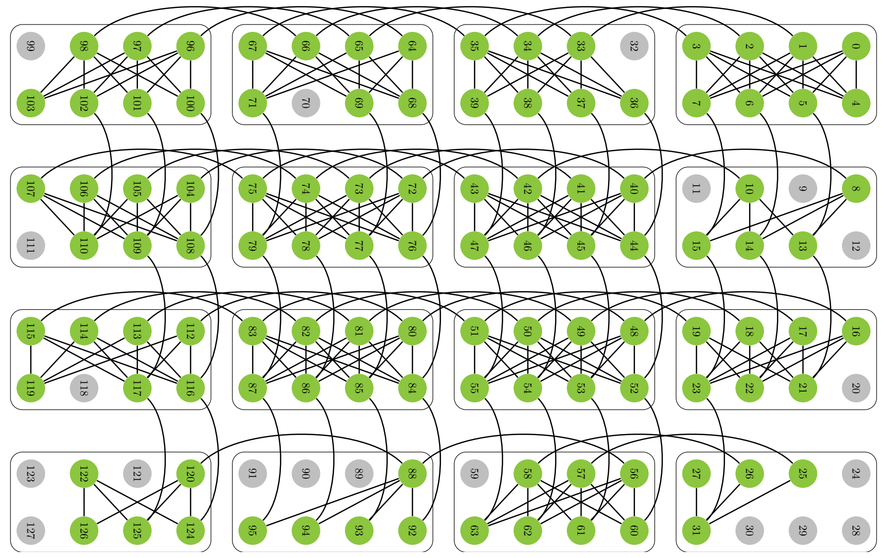
Quantum Algorithms

Cryptography

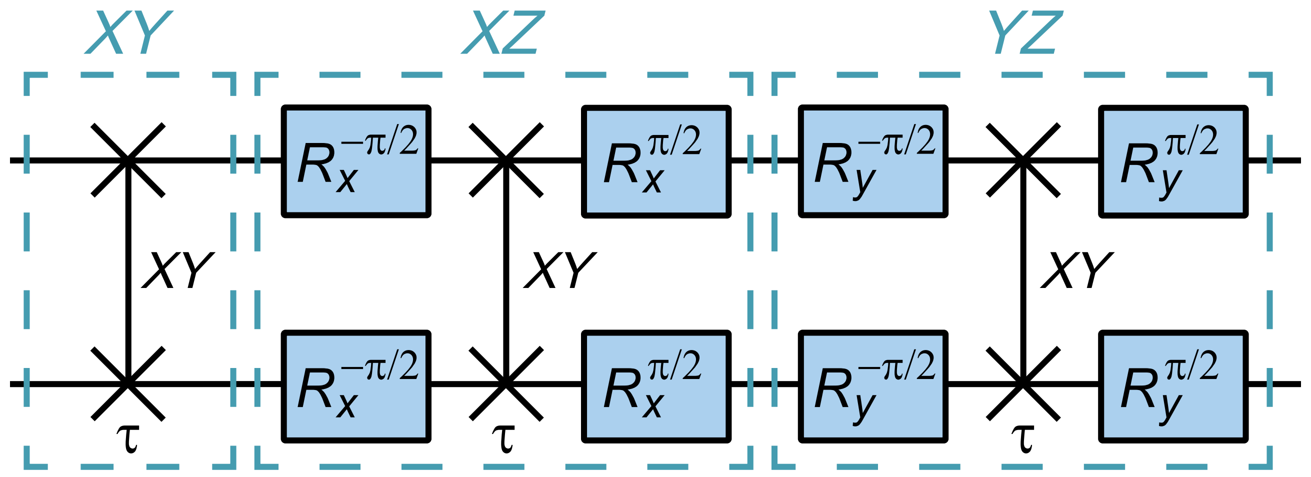
Search

$$A\,|x\rangle = |b\rangle$$

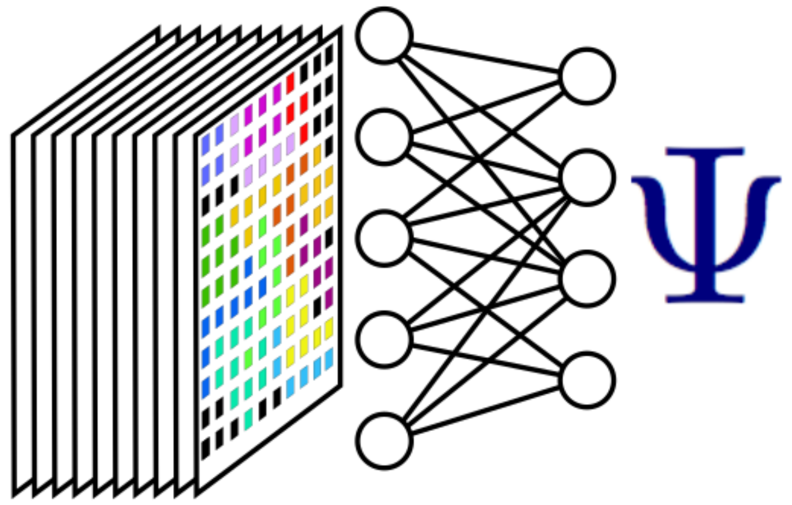
Linear Algebra



Quantum Annealing and Optimization



Quantum Simulation



Quantum Machine Learning

Quantum Algorithms

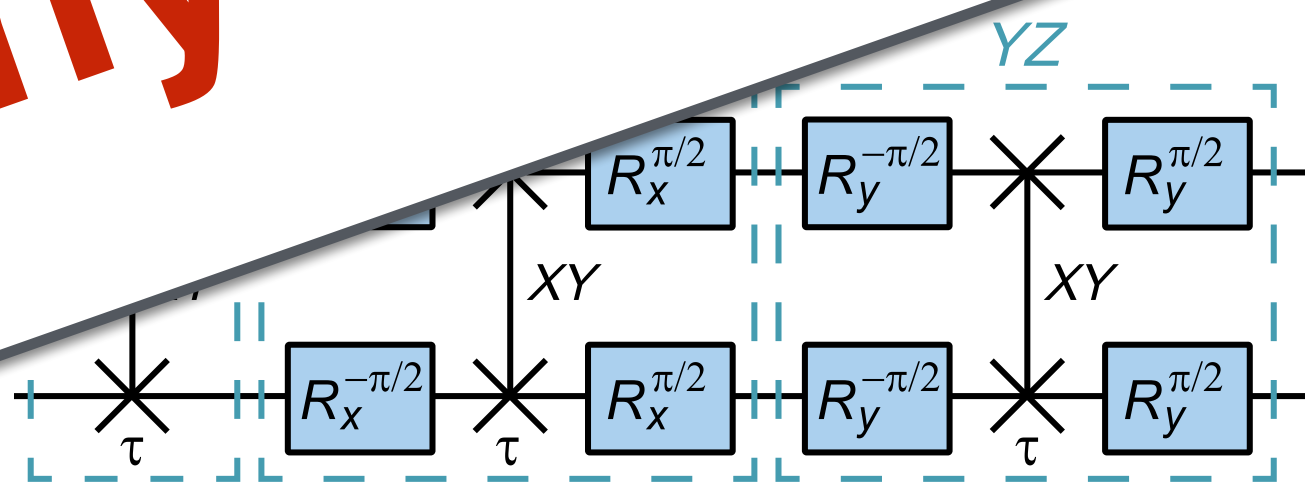
Cryptography

S

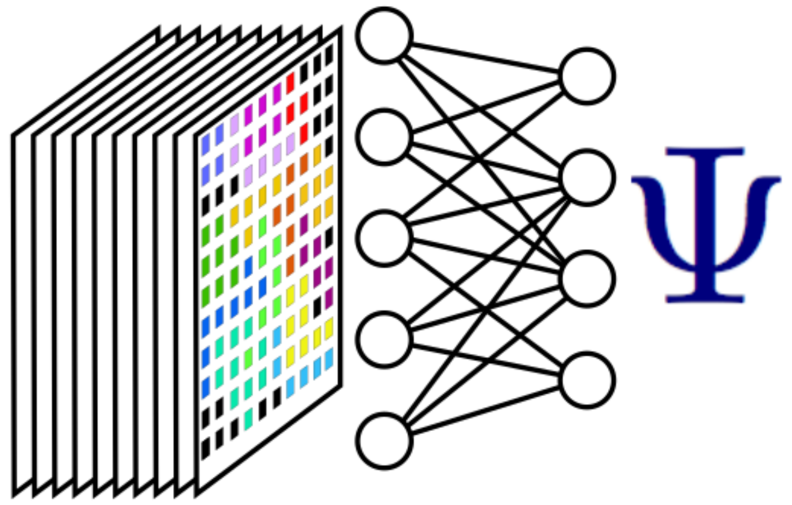
gebra

Many caveats …

tum Annealing
and Optimization

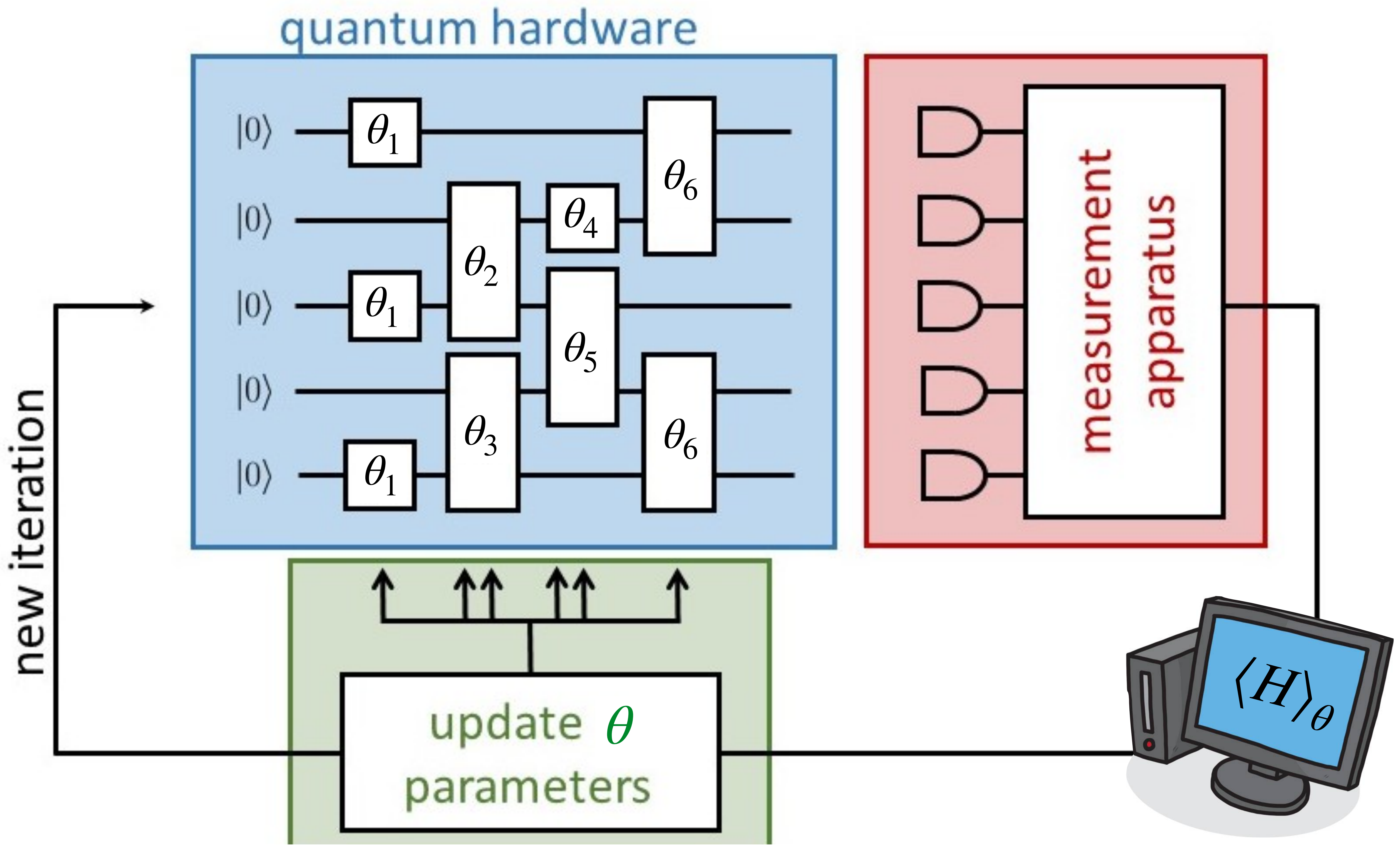


Quantum
Simulation



Quantum
Machine Learning

Variational quantum eigensolver

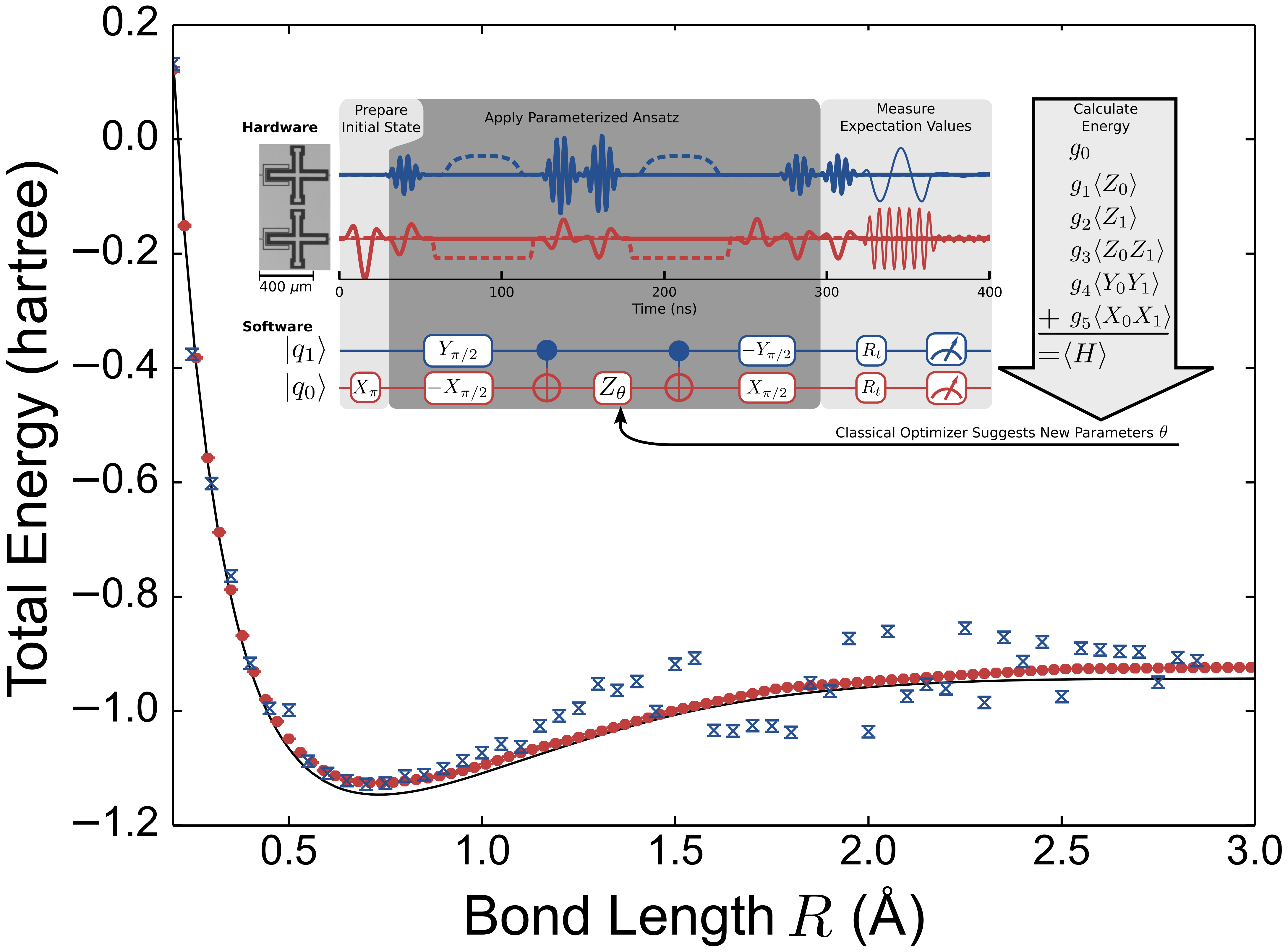


Quantum circuit as a variational ansatz

Peruzzo et al, Nat. Comm. '13

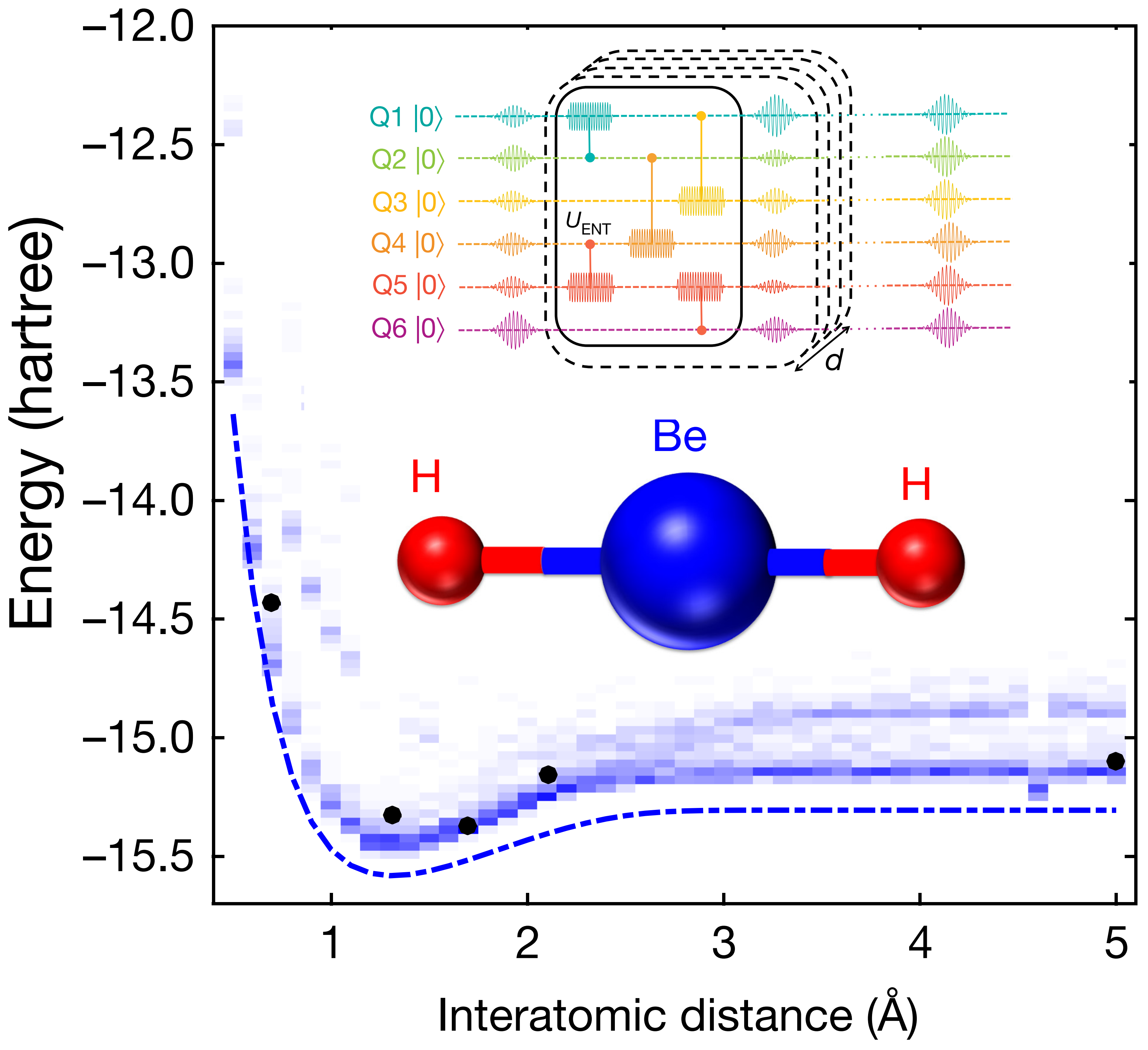
VQE on actual quantum devices

H_2 molecule with 2 qubits



Google PRX '16

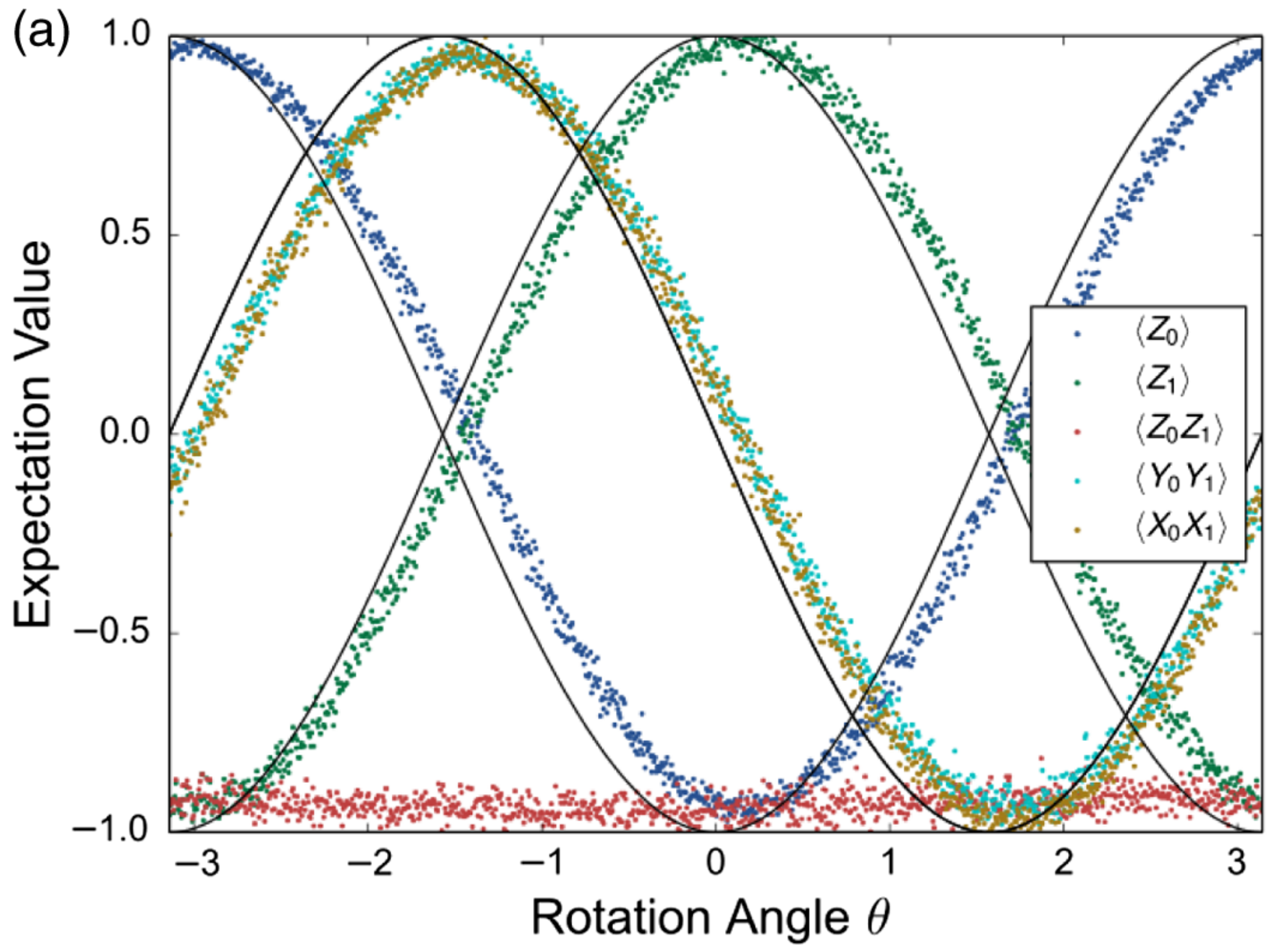
BeH_2 molecule with 6 qubits



IBM Nature '17

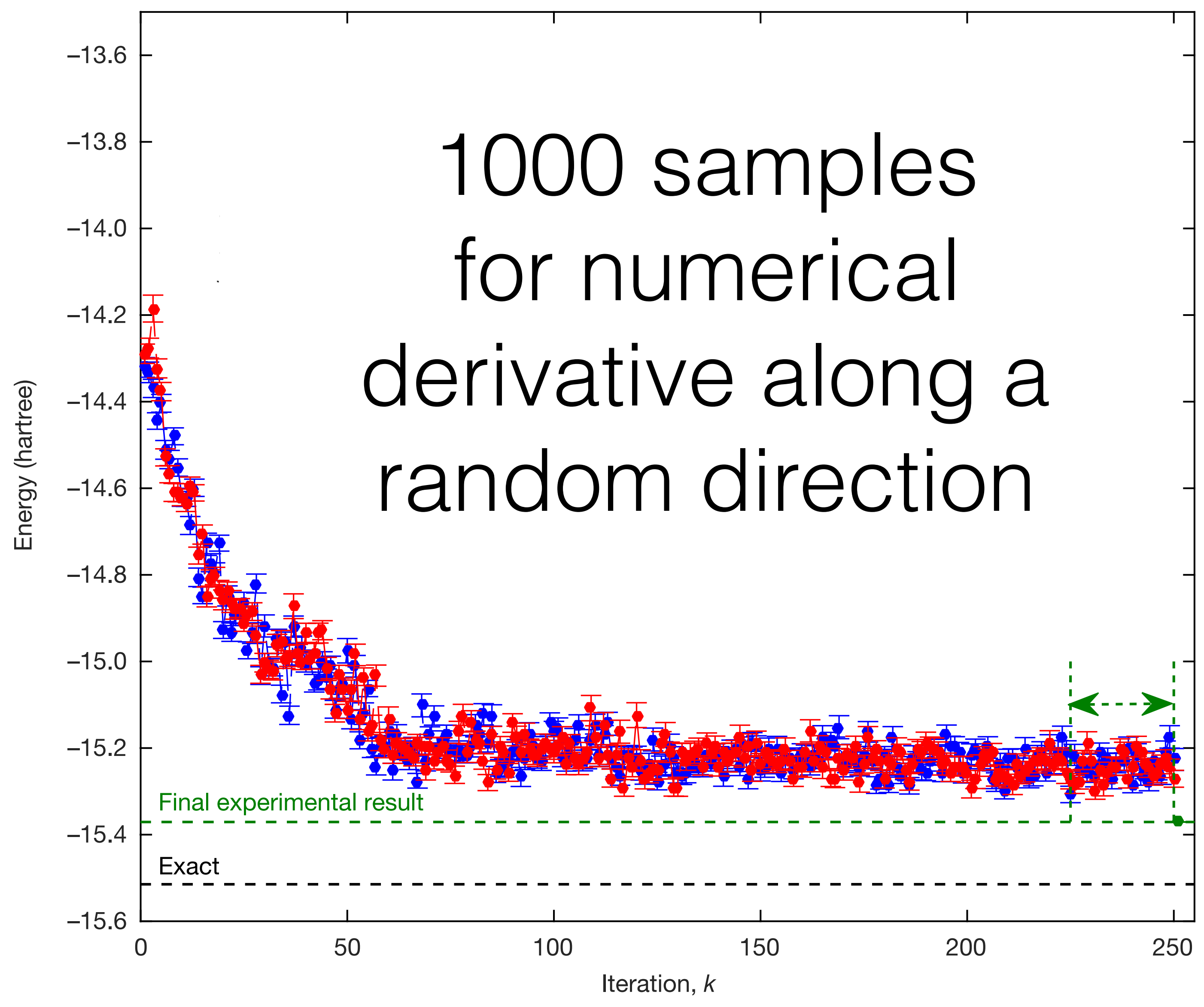
Optimize the quantum circuit

Scan 1000 values of the single variational parameter



Google PRX '16

Stochastic gradient descend with numerical derivative

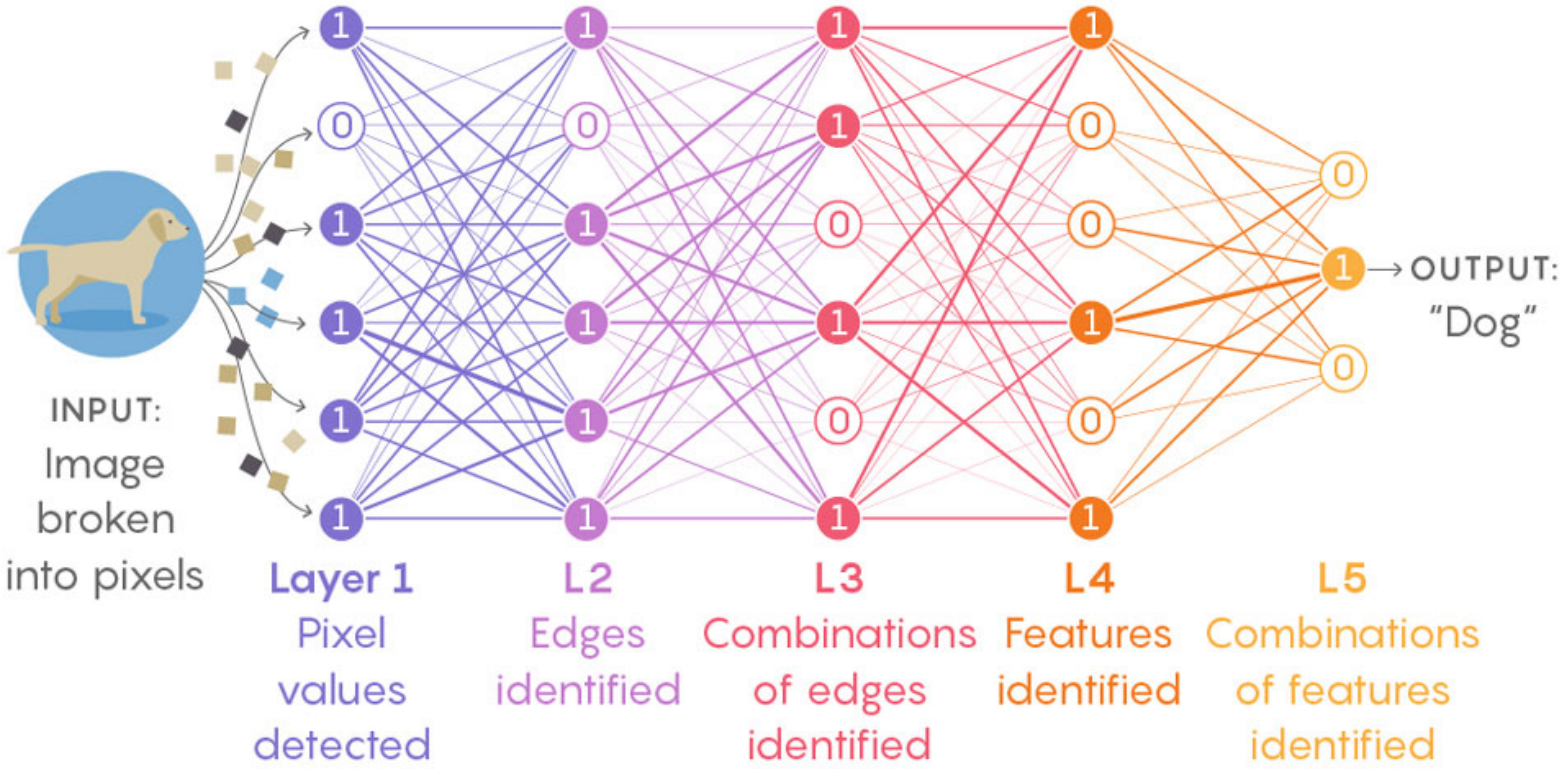


1000 samples for numerical derivative along a random direction

IBM Nature '17

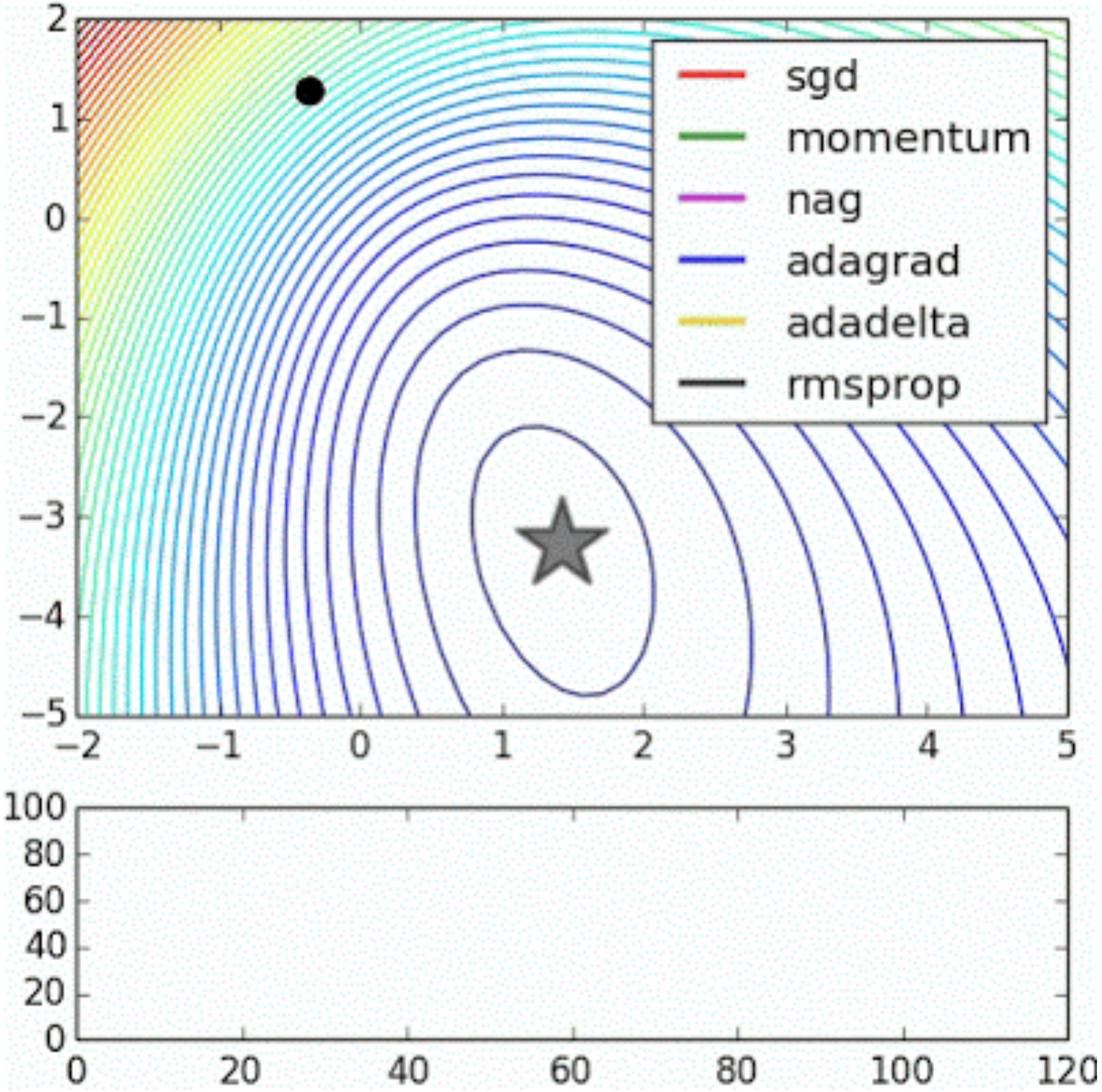
These optimization schemes do not scale to higher dimensions

The engine of deep learning



Compose differentiable components to form a program e.g. a neural network, then optimize it with gradients

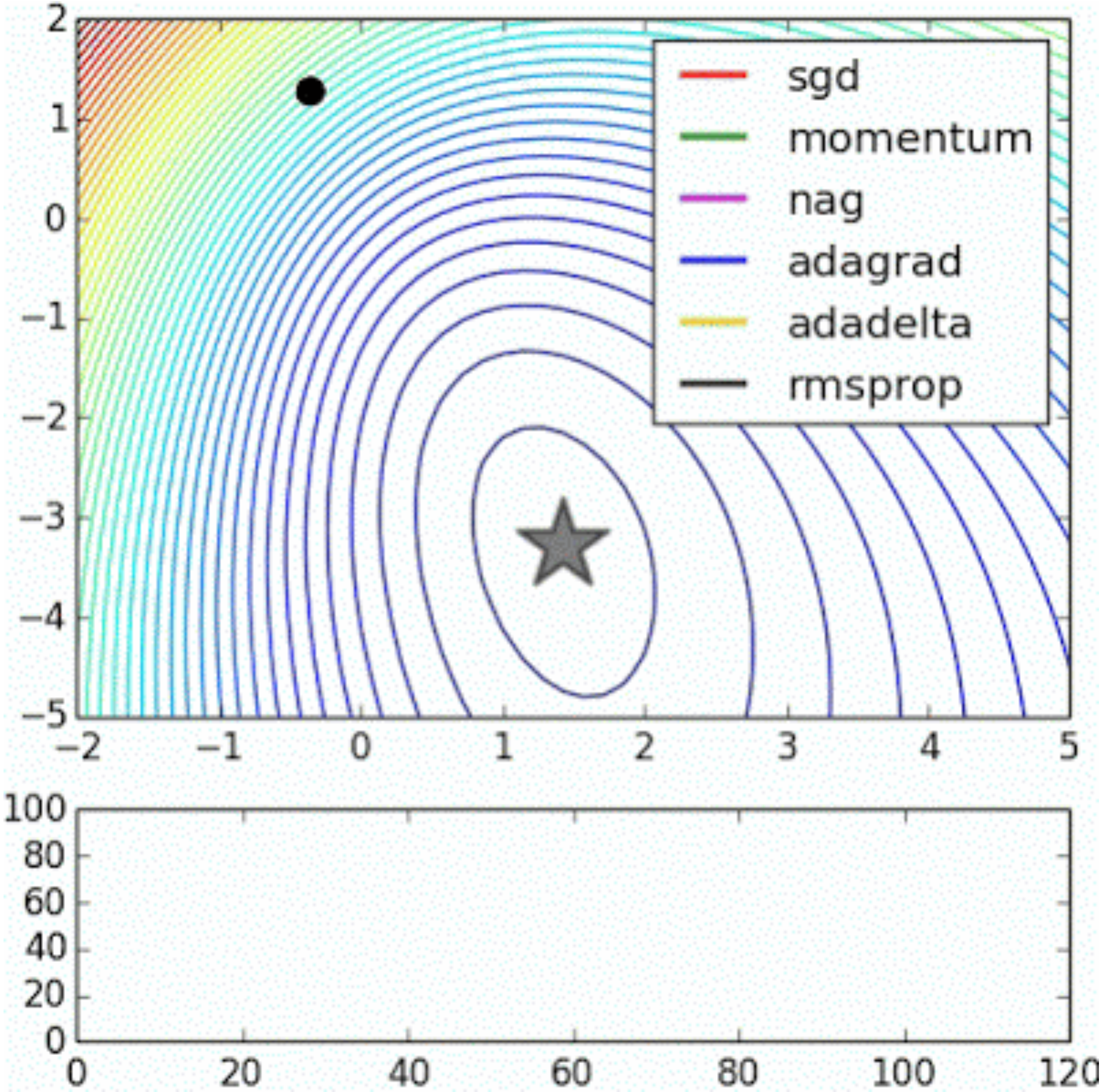
Optimization with noisy gradients



Ruder, 1609.04747

VQE encounters the "same type" of stochastic optimization in deep learning

Optimization with noisy gradients



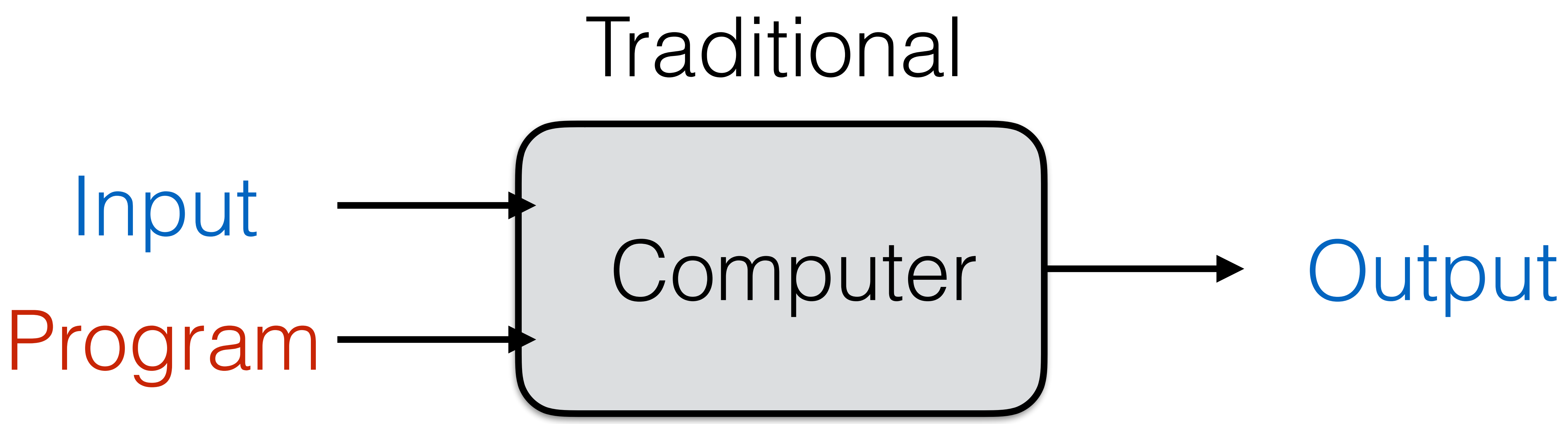
Ruder, 1609.04747

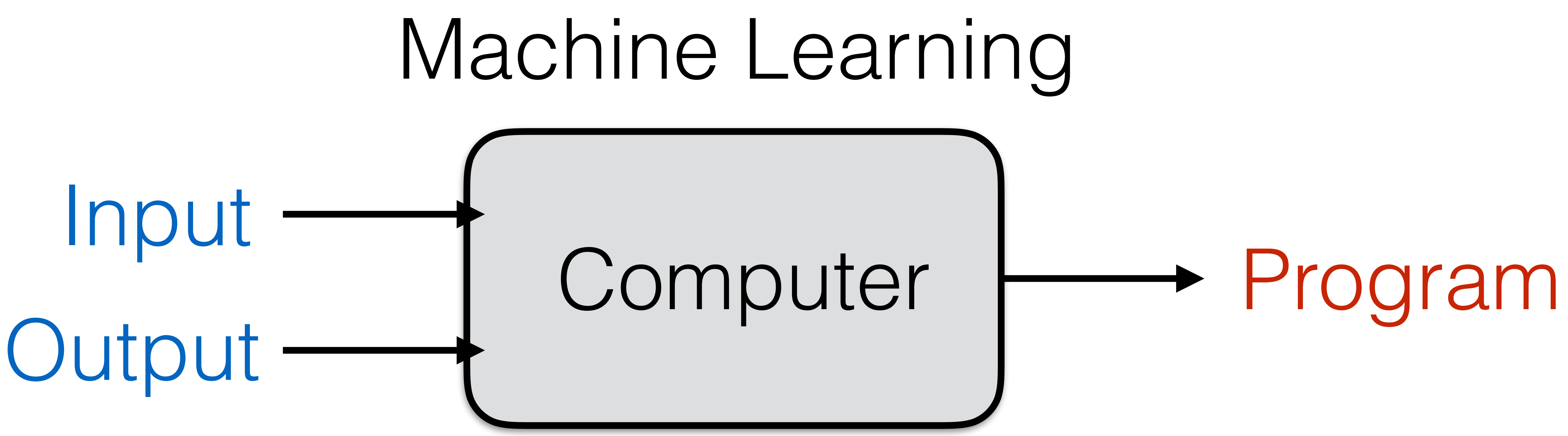
VQE encounters the “same type” of stochastic optimization in deep learning

Differentiable Programming Quantum Circuits

Neural Nets ↔ Probabilistic Graphical Models ↔ Tensor Nets ↔ Quantum Circuits

Differentiable Programming

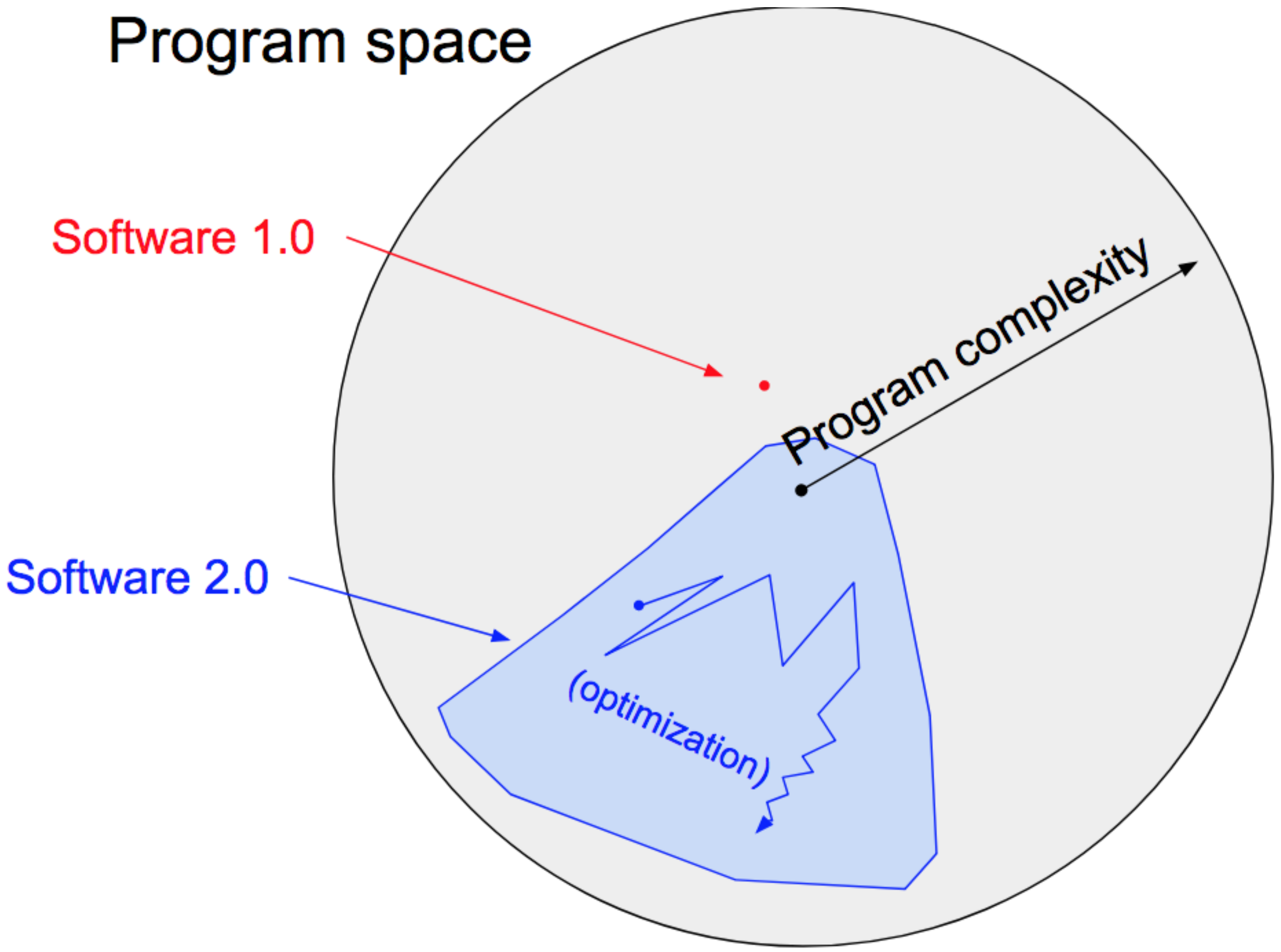




Andrej Karpathy

Director of AI at Tesla. Previously Research Scientist at OpenAI and PhD student at Stanford. I like to train deep neural nets on large datasets.

https://medium.com/@karpathy/software-2-0-a64152b37c35



Writing software 2.0 by gradient search in the program space

Differentiable Programming

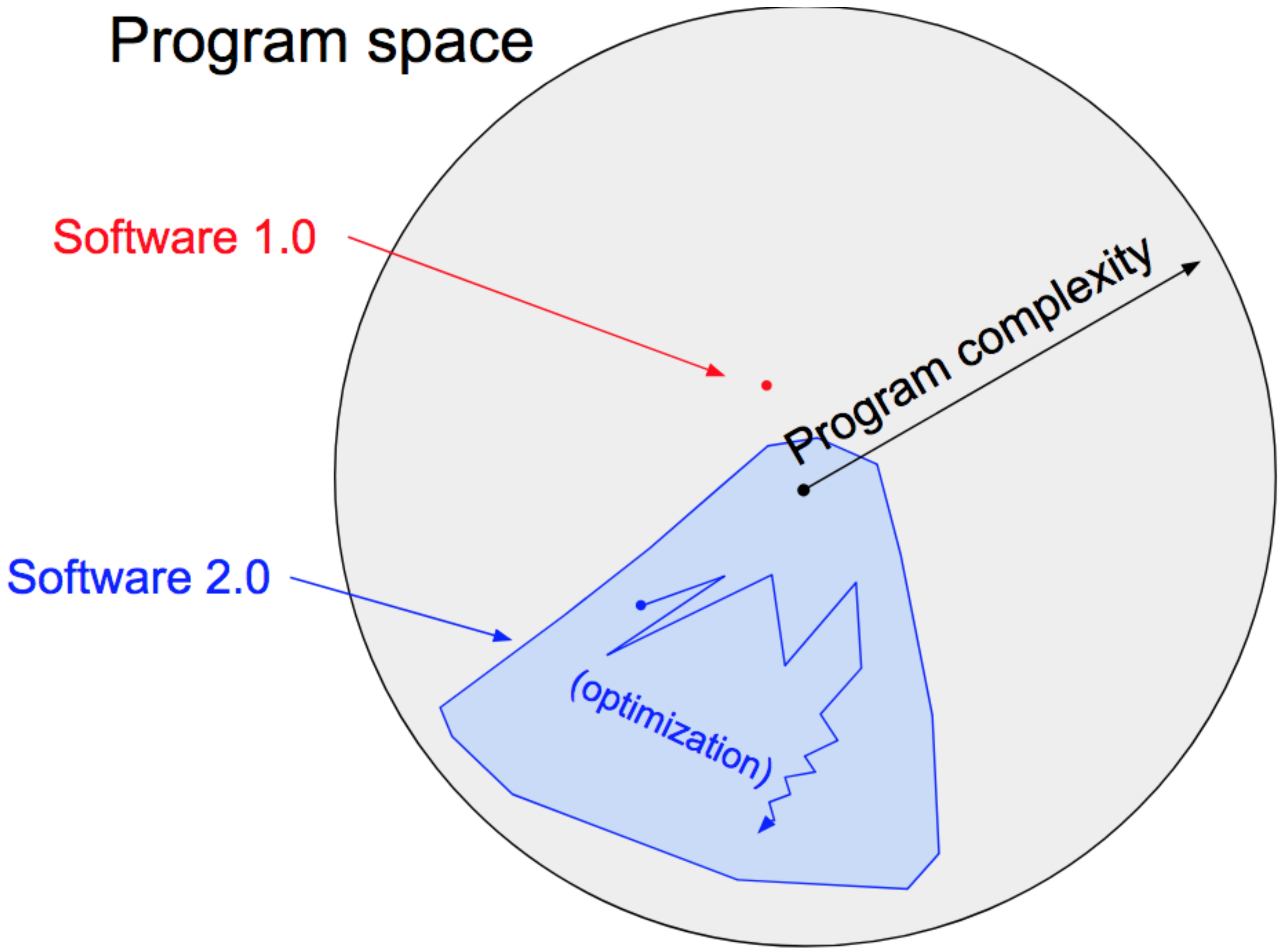
Benefits of Software 2.0

- Computationally homogeneous
- Simple to bake into silicon
- Constant running time
- Constant memory usage
- Highly portable & agile
- Modules can meld into an optimal whole
- **Better than humans**

Andrej Karpathy

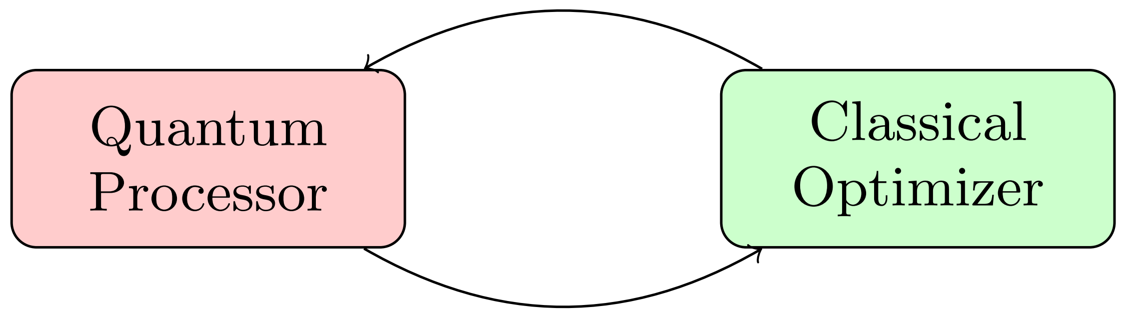
Director of AI at Tesla. Previously Research Scientist at OpenAI and PhD student at Stanford. I like to train deep neural nets on large datasets.

https://medium.com/@karpathy/software-2-0-a64152b37c35



Writing software 2.0 by gradient search in the program space

Differentiable Quantum Programming



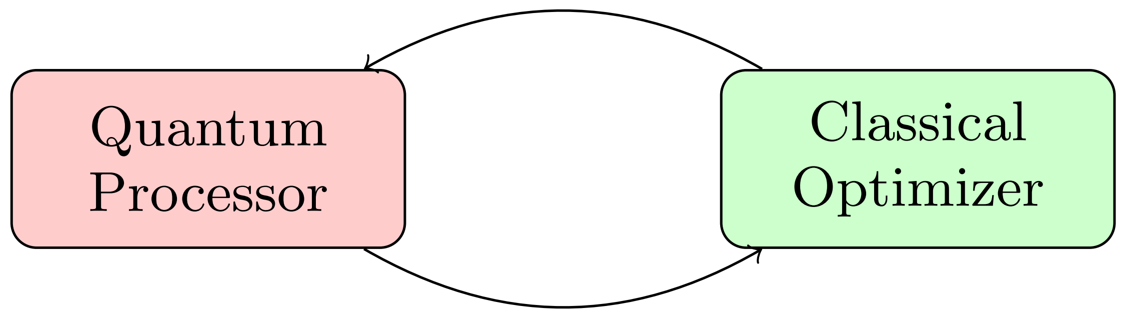
It is a paradigm beyond quantum-classical hybrid

- Variational quantum eigensovler (VQE)
- Quantum circuit Born machine (QCBM)
- Quantum approximate optimization algorithm (QAOA)
- Quantum pattern recognition

…

Quantum circuit classifier	Farhi, Neven, 1802.06002 Havlicek et al, 1804.11326
TNS inspired circuit architecture	Huggins, Patel, Whaley, Stoudenmire, 1803.11537
VQE with fewer qubits	Liu, Zhang, Wan, LW, 1902.02663
Quantum generative model	Gao, Zhang, Duan, 1711.02038
Quantum adversarial training	Dallaire-Demers, Lloyd, Benedetti 1804.08641,1804.09139, 1806.00463

Differentiable Quantum Programming



It is a paradigm beyond quantum-classical hybrid

- Variational quantum eigensovler (VQE)
- Quantum circuit Born machine (QCBM)
- Quantum approximate optimization algorithm (QAOA)
- Quantum pattern recognition

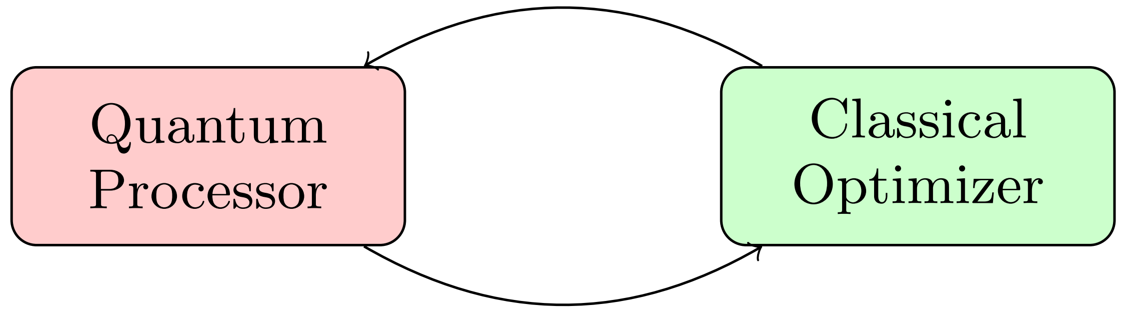
…

Near term：
What can we do with noisy circuits of limited depth ?

Long term:
Are we really good at programing quantum computers ?

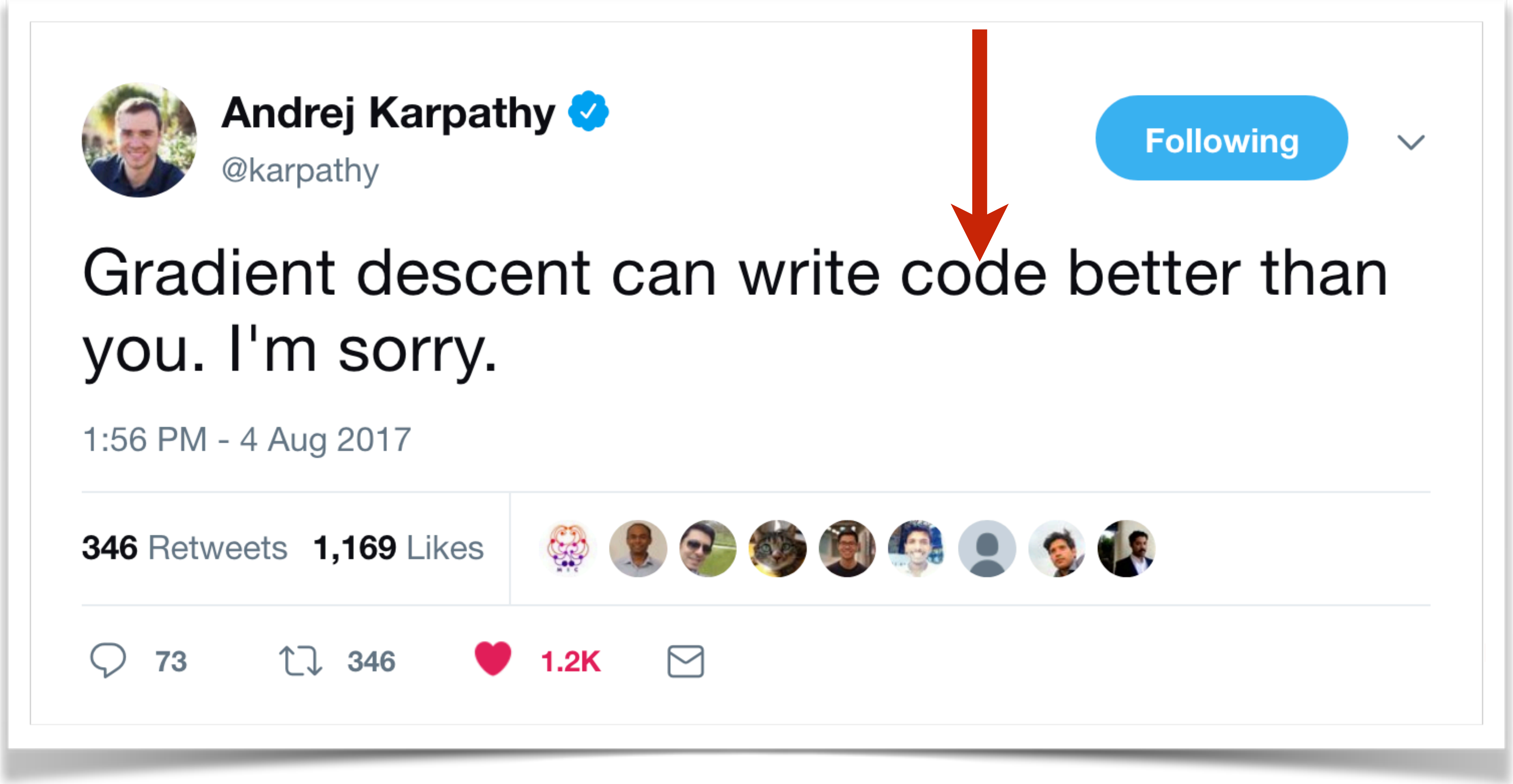
Quantum circuit classifier	Farhi, Neven, 1802.06002 Havlicek et al, 1804.11326
TNS inspired circuit architecture	Huggins, Patel, Whaley, Stoudenmire, 1803.11537
VQE with fewer qubits	Liu, Zhang, Wan, LW, 1902.02663
Quantum generative model	Gao, Zhang, Duan, 1711.02038
Quantum adversarial training	Dallaire-Demers, Lloyd, Benedetti 1804.08641,1804.09139, 1806.00463

Differentiable Quantum Programming



It is a paradigm beyond quantum-classical hybrid

Quantum code

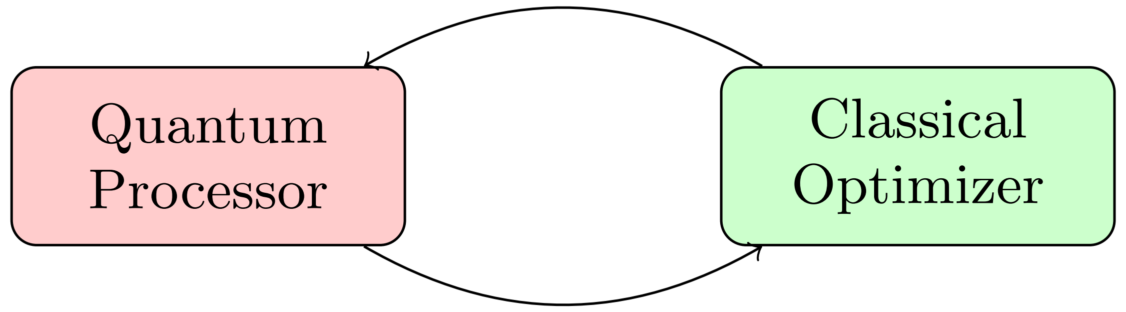


- Variational quantum eigensovler (VQE)
- Quantum circuit Born machine (QCBM)
- Quantum approximate optimization algorithm (QAOA)
- Quantum pattern recognition

…

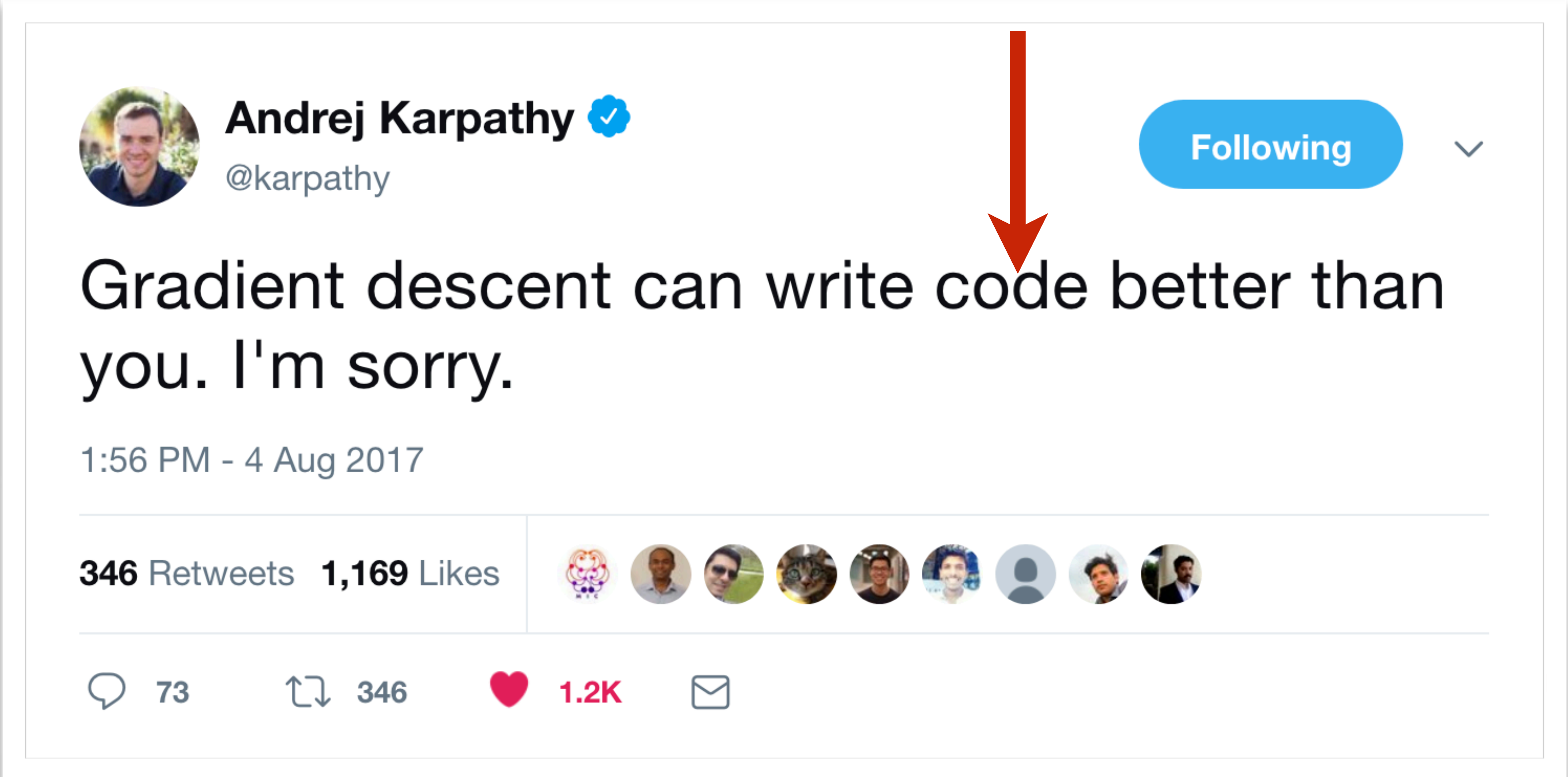
Quantum circuit classifier	Farhi, Neven, 1802.06002 Havlicek et al, 1804.11326
TNS inspired circuit architecture	Huggins, Patel, Whaley, Stoudenmire, 1803.11537
VQE with fewer qubits	Liu, Zhang, Wan, LW, 1902.02663
Quantum generative model	Gao, Zhang, Duan, 1711.02038
Quantum adversarial training	Dallaire-Demers, Lloyd, Benedetti 1804.08641,1804.09139, 1806.00463

Differentiable Quantum Programming



It is a paradigm beyond quantum-classical hybrid

Quantum code



- Variational quantum eigensovler (VQE)
- Quantum circuit Born machine (QCBM)
- Quantum approximate optimization algorithm (QAOA)
- Quantum pattern recognition

…

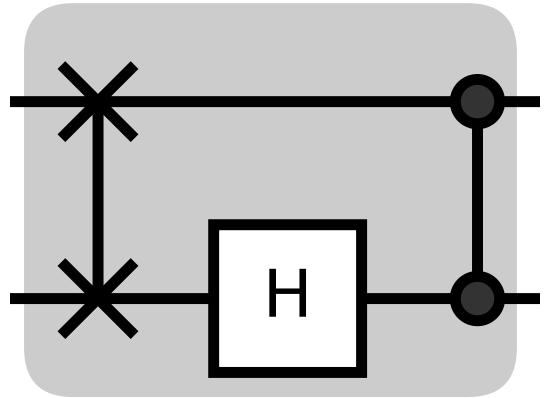
Quantum circuit
TNS inspired circ
VQE with fewer q
Quantum genera
Quantum advers

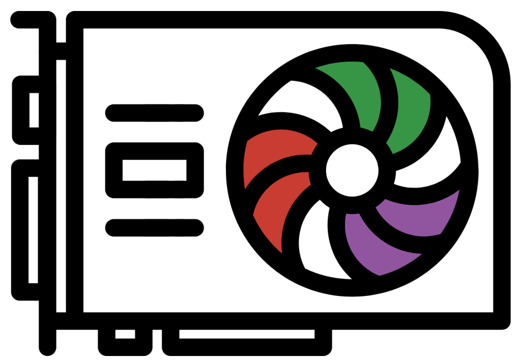
Quantum Software 2.0

1806.00463

Be prepared for Quantum Software 2.0

https://yaoquantum.org/

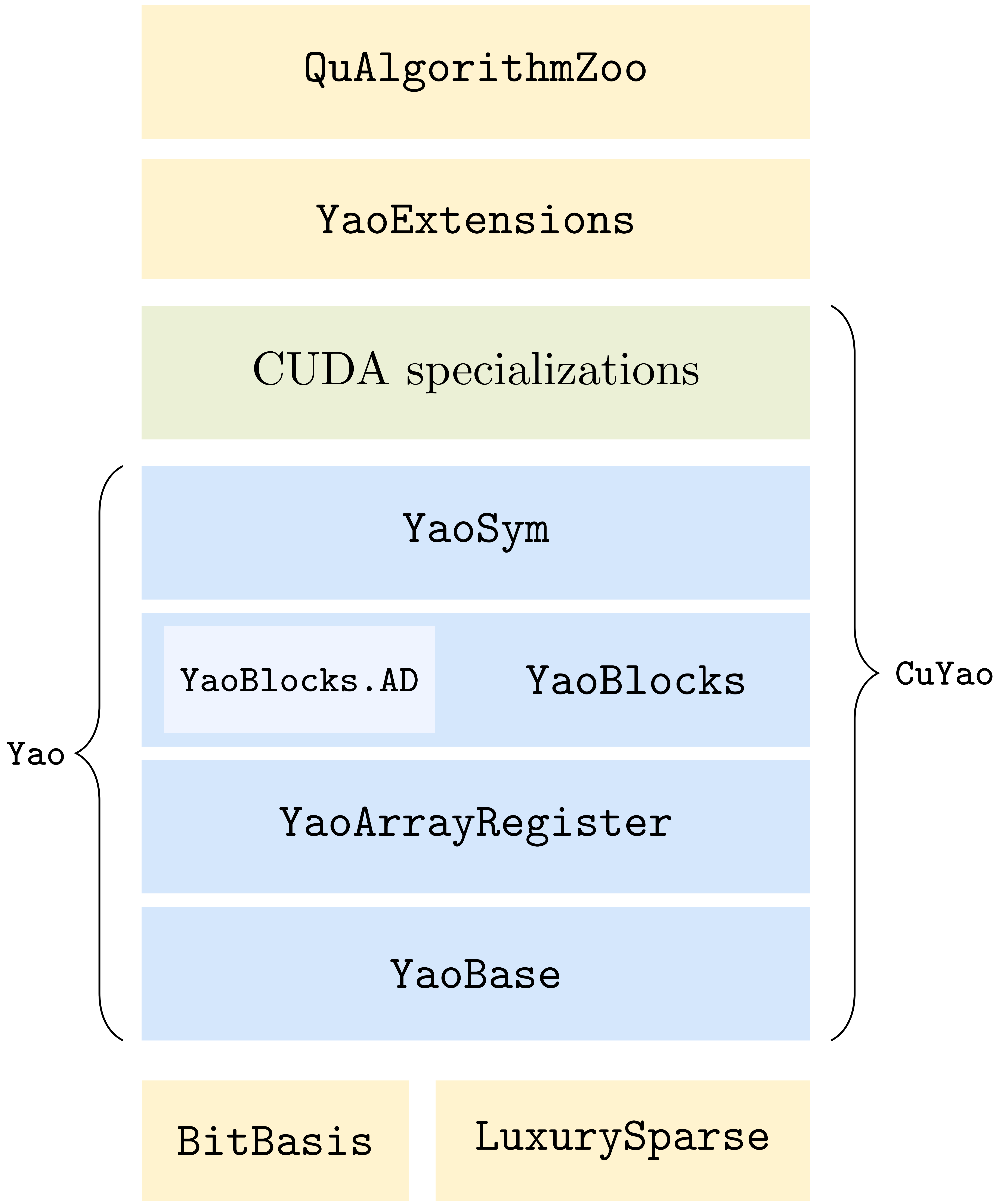
\+ 

\+  =

Xiu-Zhe Luo (IOP, CAS → Waterloo & PI)
Jin-Guo Liu (IOP, CAS → Harvard)

Features:

- Differentiable programming quantum circuits
- Batched quantum register with GPU acceleration
- Quantum block intermediate representation

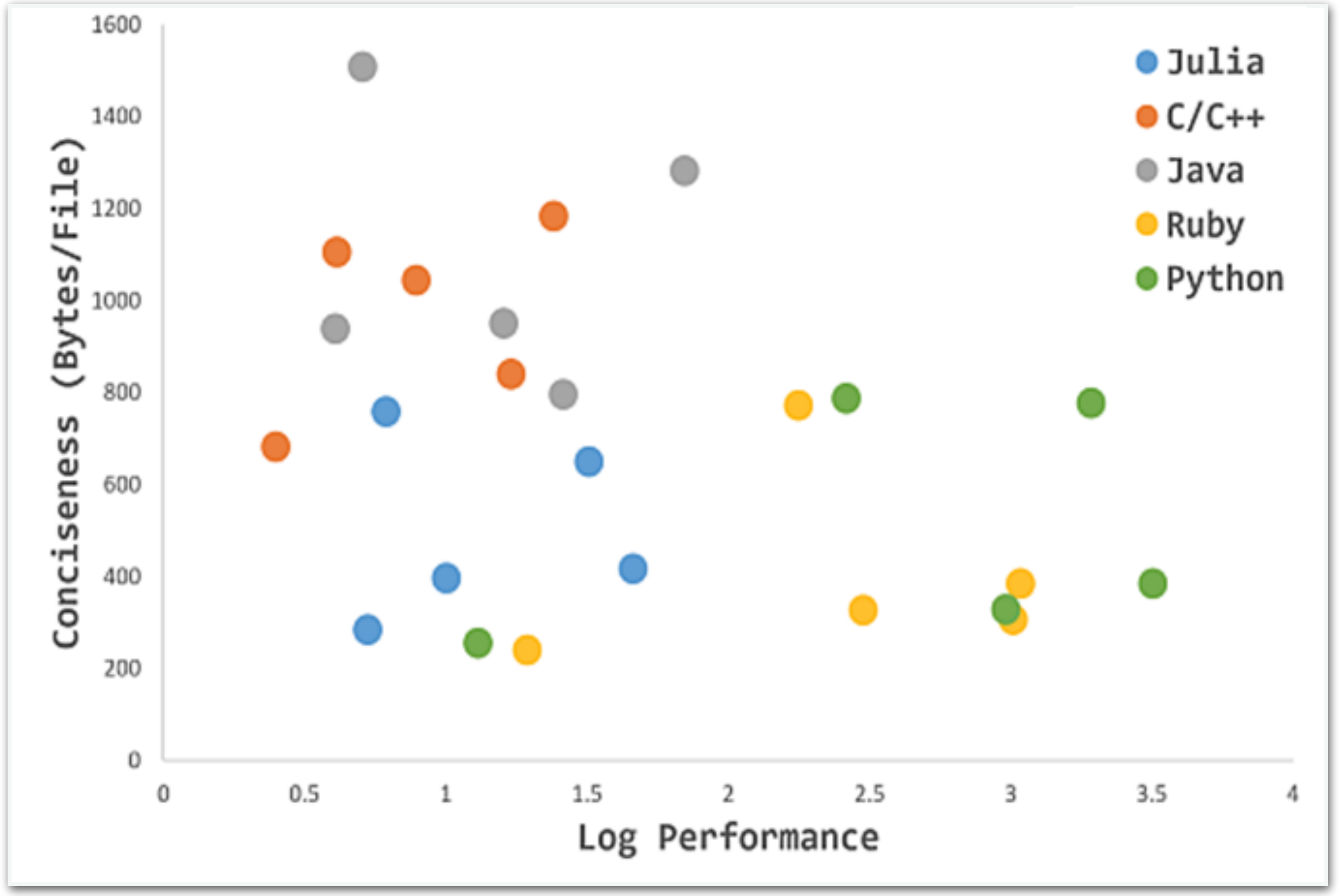


Stacks of Yao

https://github.com/QuantumBFS

Why Julia ?

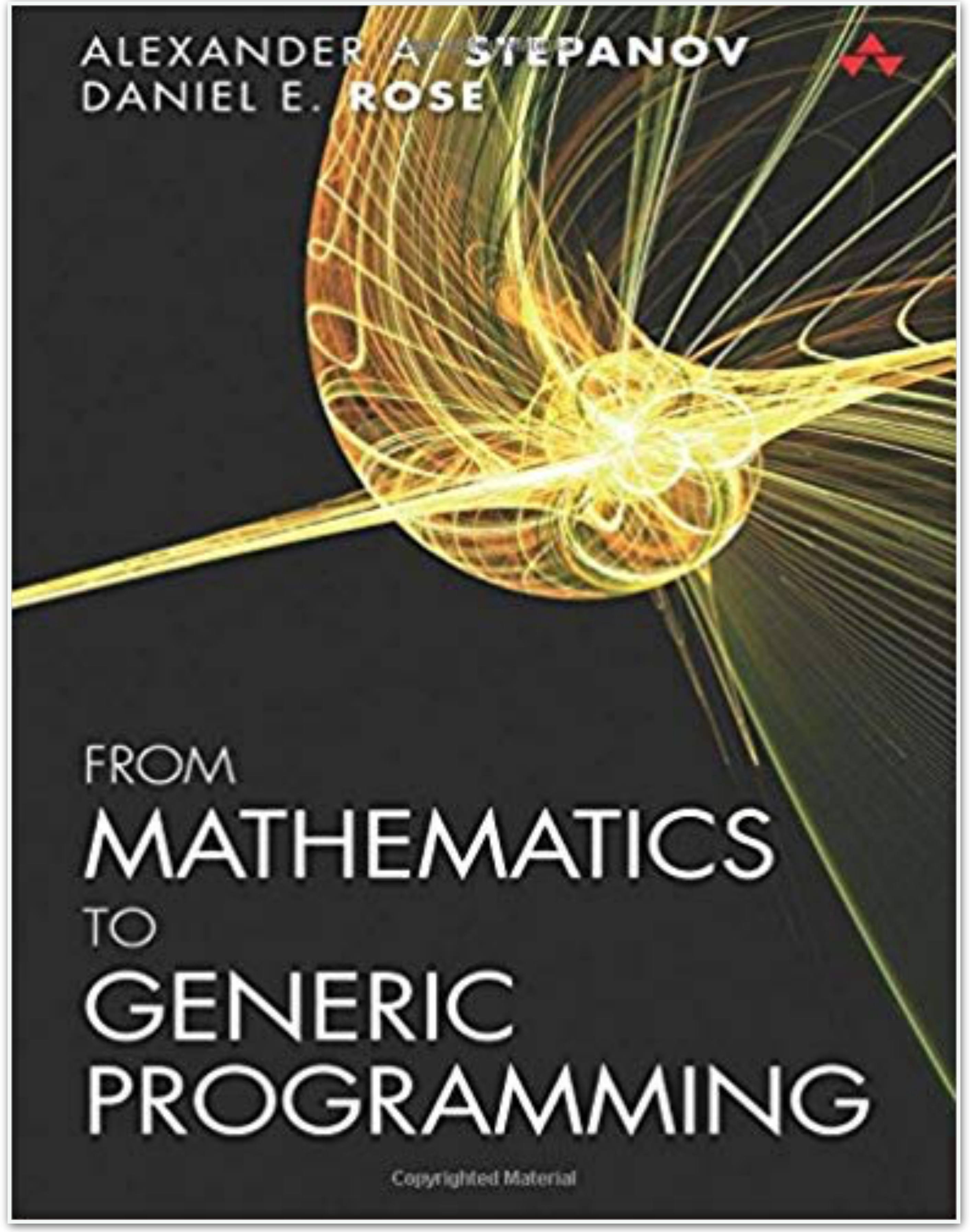
- Julia is fast!
- Generic programming (type system and multiple dispatch)
- The future of technical computing



http://ljuug.org

Why Julia ?

- Julia is fast!
- Generic programming (type system and multiple dispatch)
- The future of technical computing

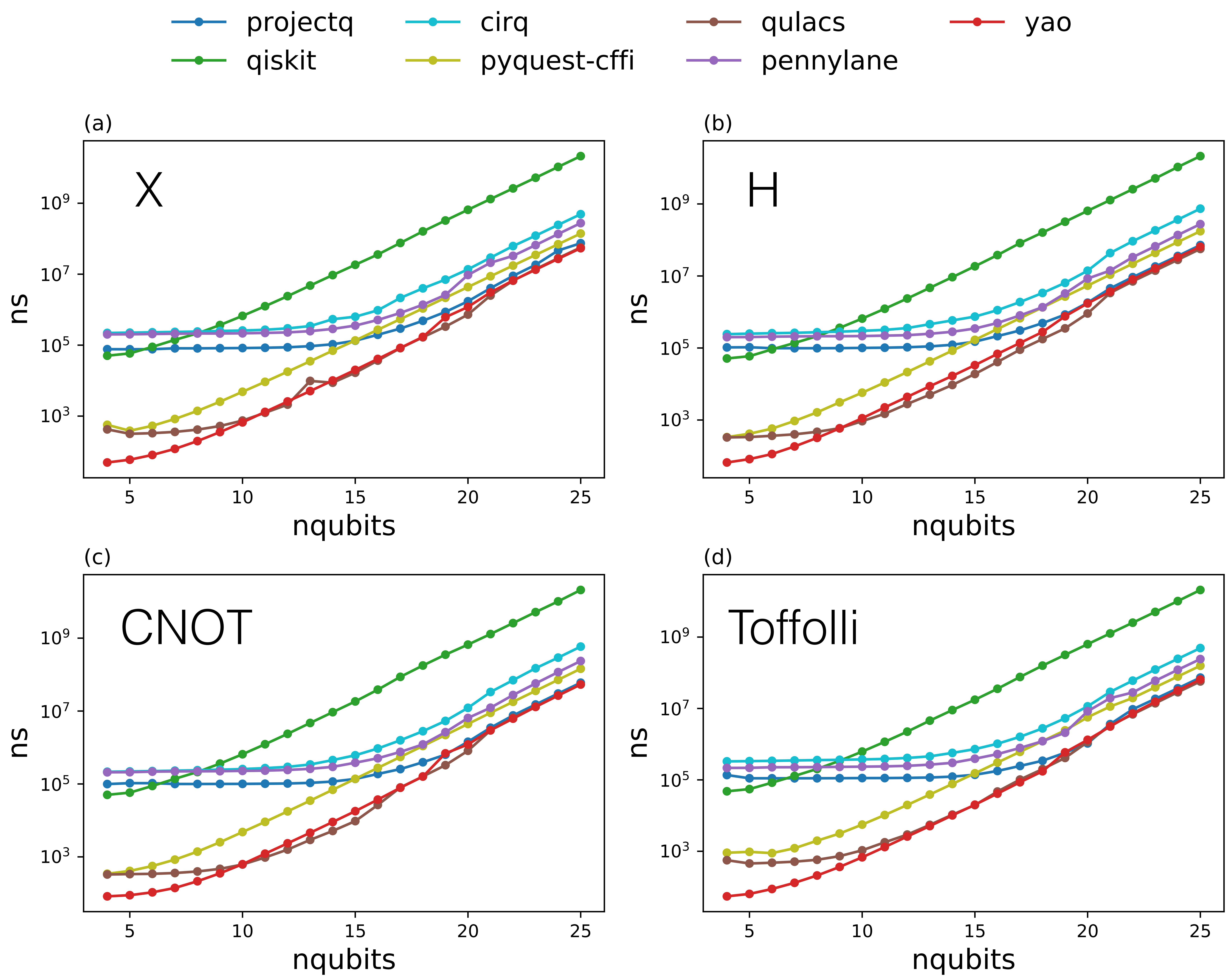


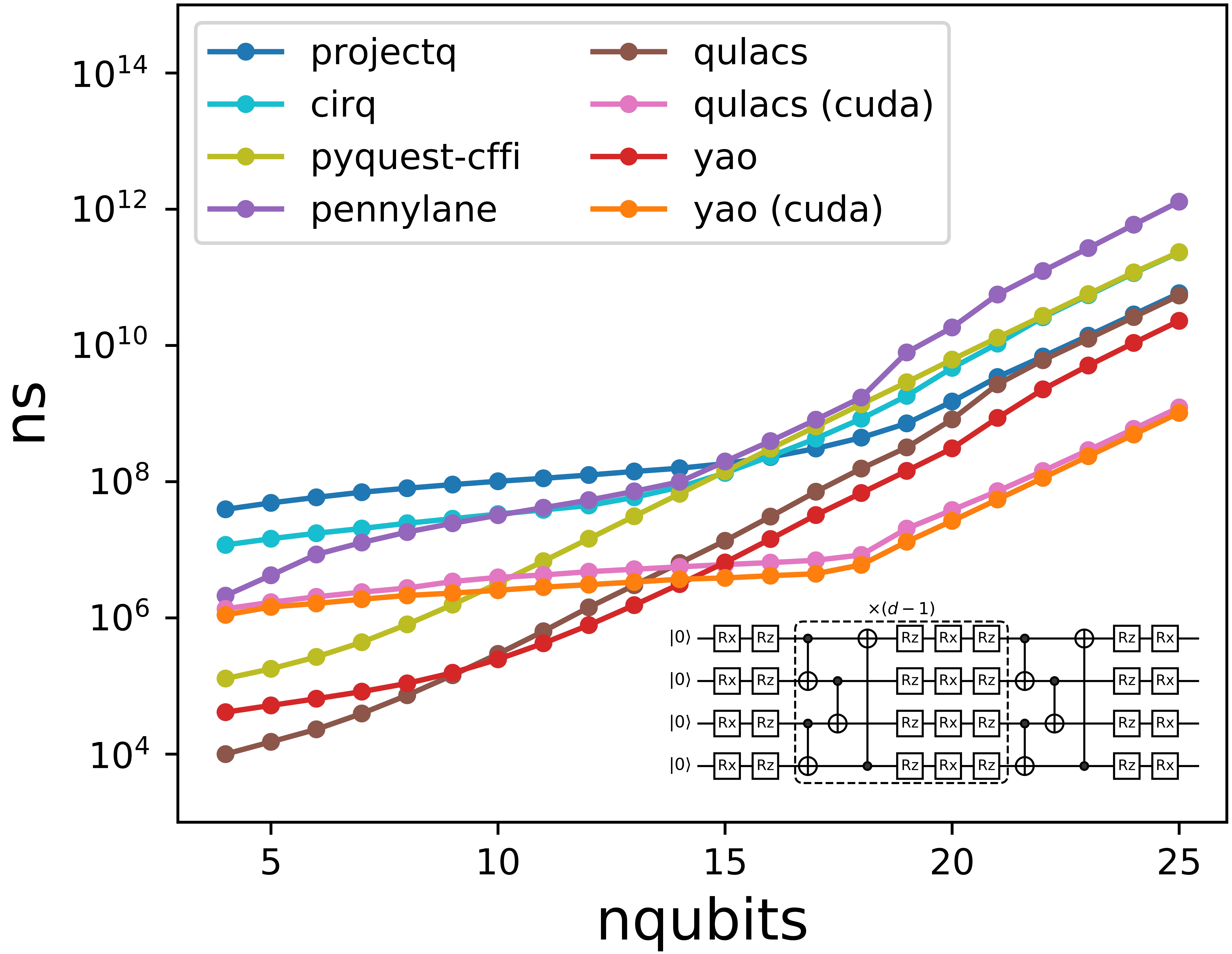
Why Julia ?

- Julia is fast!
- Generic programming (type system and multiple dispatch)
- Future of technical computing

Demo 1

https://github.com/wangleiphy/YaoTutorial

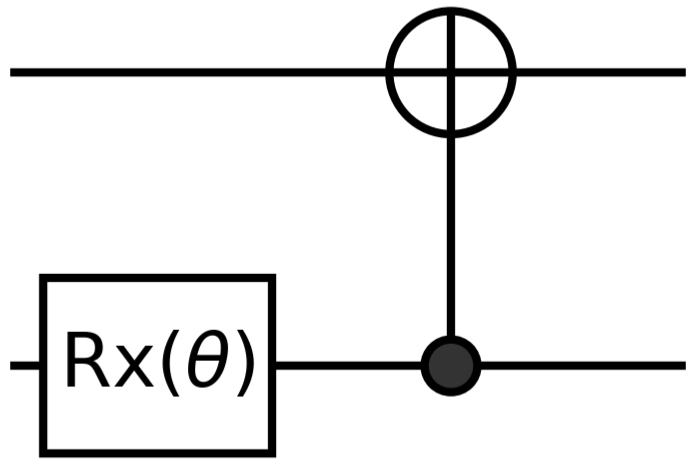

projectq
qiskit
cirq
pyquest-cffi
qulacs
pennylane
yao
(a)
X
(b)
H
(c)
CNOT
(d)
Toffolli
ns
nqubits
10^3
10^5
10^7
10^9
5
10
15
20
25



projectq
cirq
pyquest-cffi
pennylane
qulacs
qulacs (cuda)
yao
yao (cuda)
ns
nqubits
×(d − 1)
|0⟩
Rx
Rz

Quantum Block Intermediate Representation

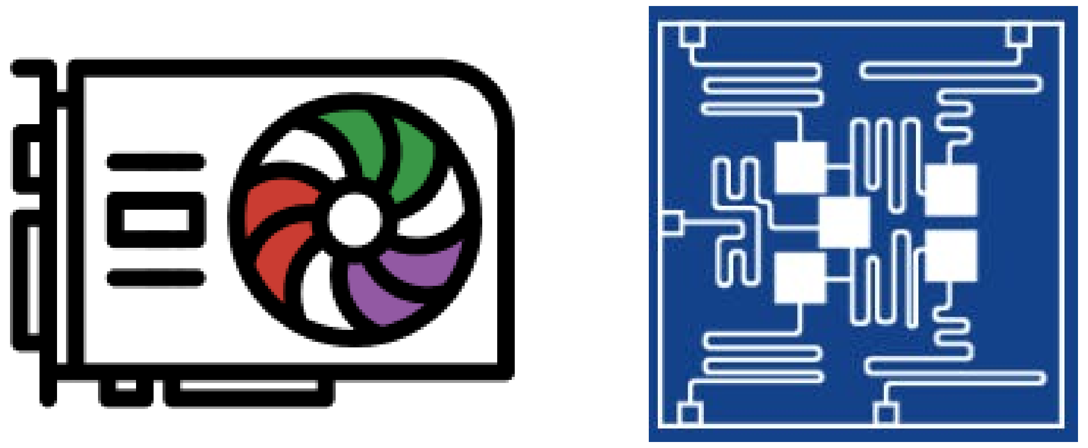
Quantum Algorithms



QBIR

```
julia> circuit = chain(2,
           put(2=>Rx(θ)),
           control(2, 1=>X))
nqubits: 2
chain
├─ put on (2)
│  └─ rot(XGate, θ)
└─ control(2)
   └─ (1,) X gate
```

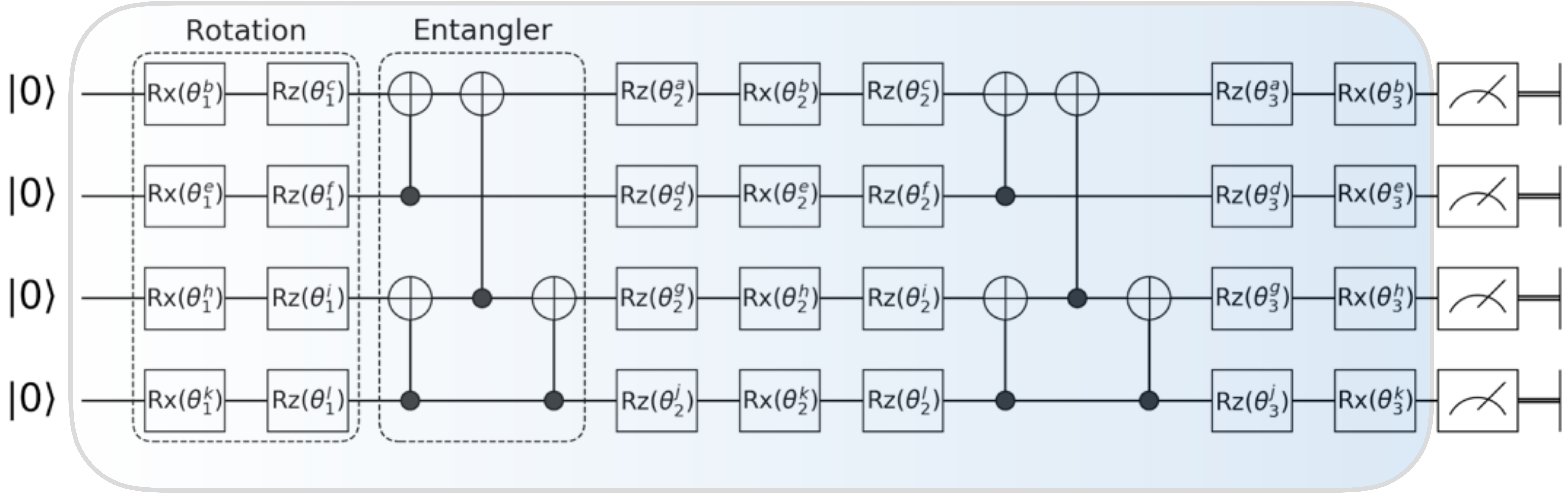
Quantum Registers



Demo 2

https://github.com/wangleiphy/YaoTutorial

Differentiable[1] quantum circuits



Write your simulator as a machine learning model
Isn't that obvious ?

Differentiable programming tools

HIPS/autograd

theano

flux

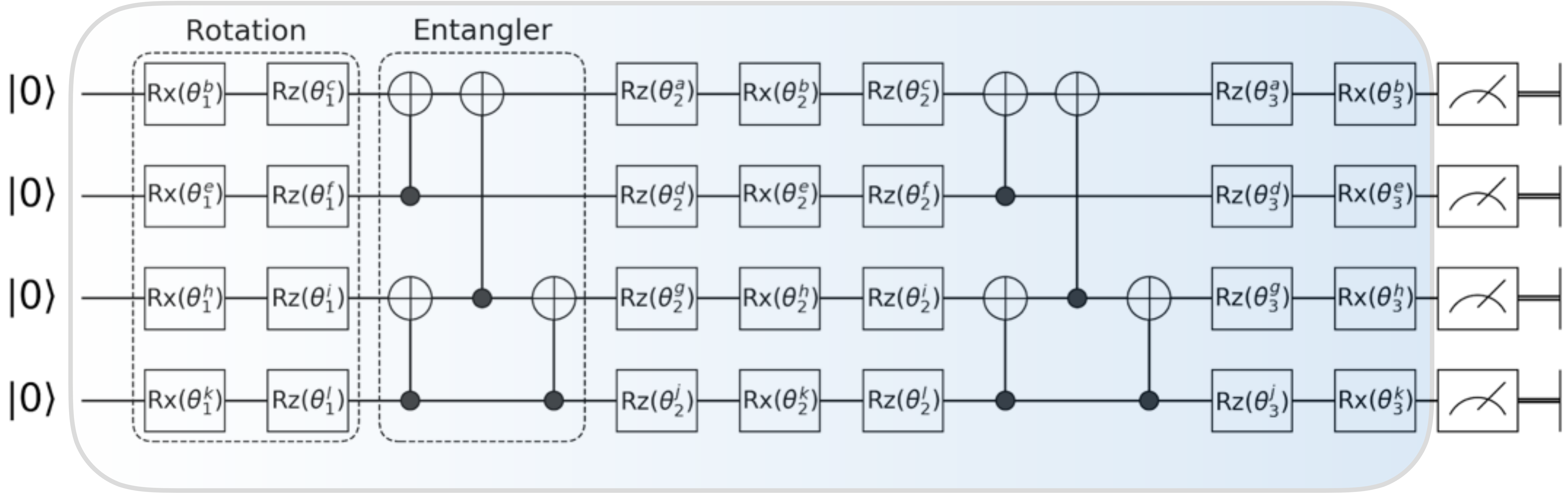
PyTorch

TensorFlow

Zygote

Keras

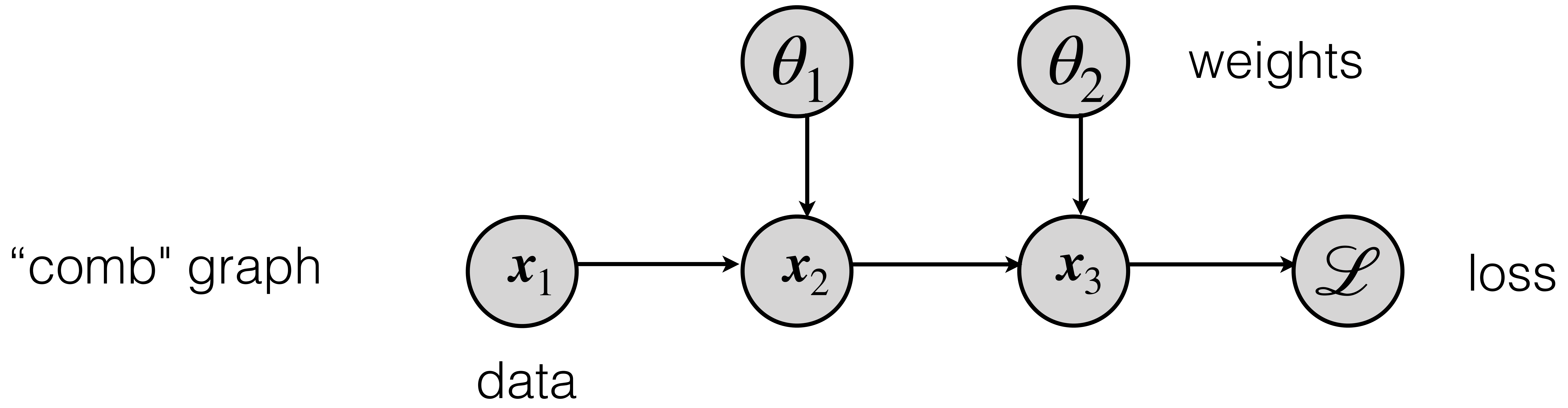
Differentiable[1] quantum circuits



Even better: quantum computing is reversible!
Backpropagation with O(1) memory in classical simulation

Reversible training of neural nets Gomez et al, 1707.04585 Chen et al, 1806.07366

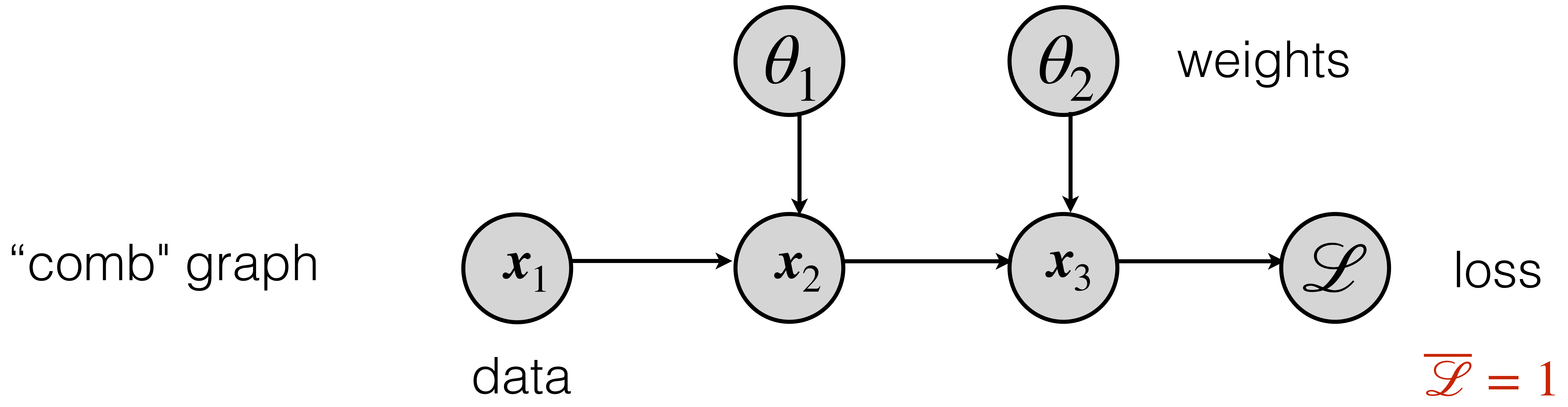
Automatic differentiation on computation graph



Define "adjoint" $\bar{x} = \frac{\partial \mathscr{L}}{\partial x}$

Pullback the adjoint through the graph

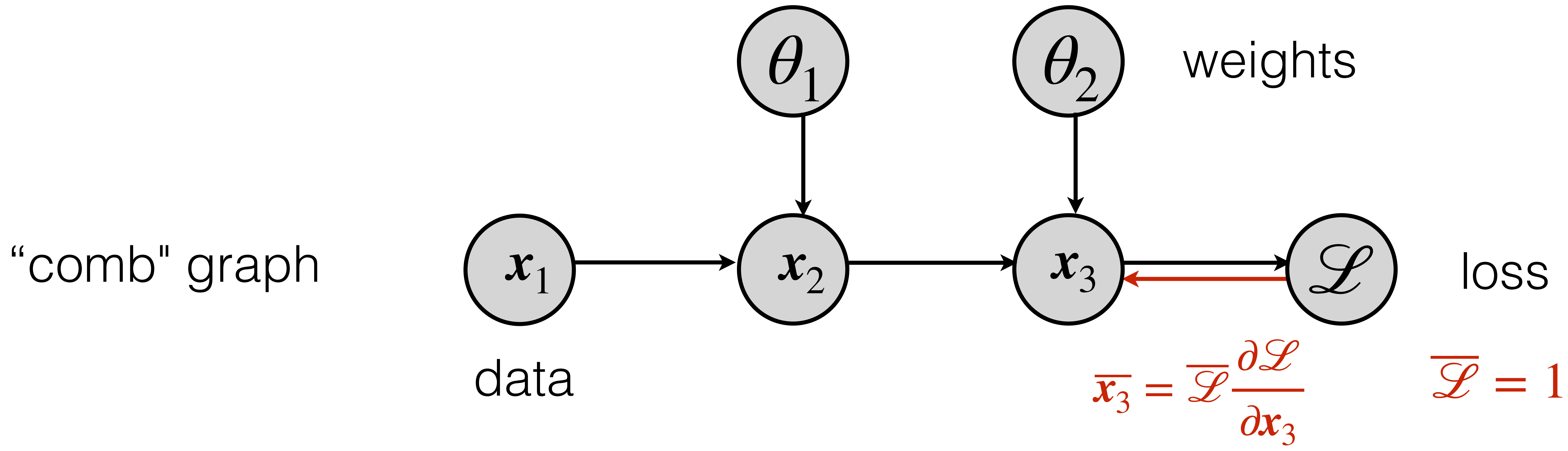
Automatic differentiation on computation graph



Define "adjoint" $\bar{x} = \frac{\partial \mathscr{L}}{\partial x}$

Pullback the adjoint through the graph

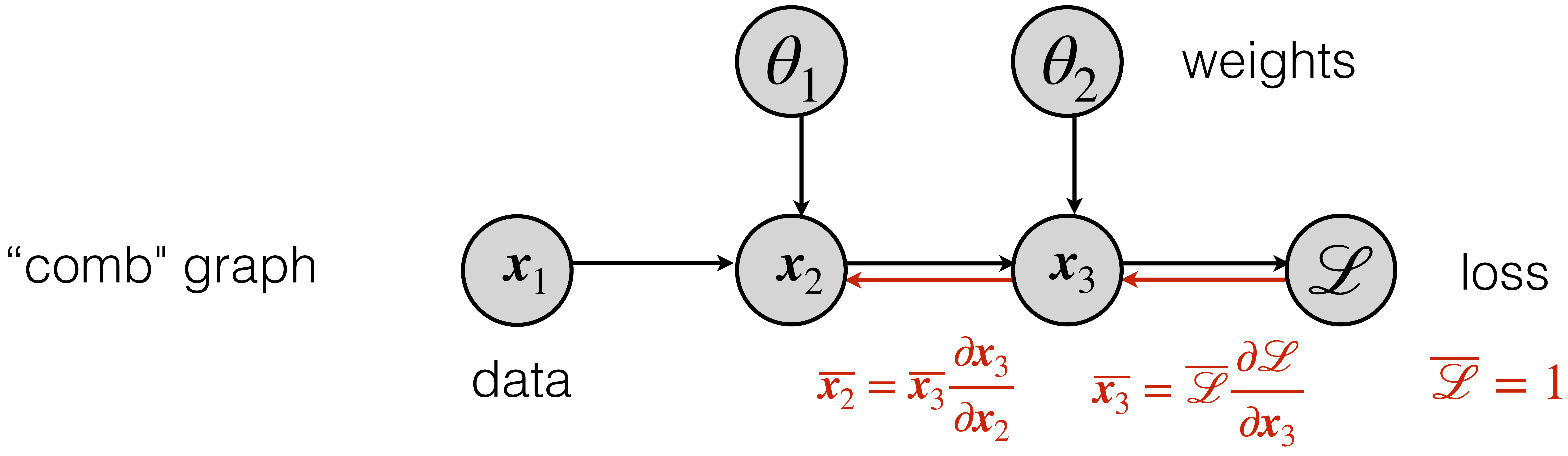
Automatic differentiation on computation graph



Define "adjoint" $\bar{x} = \frac{\partial \mathcal{L}}{\partial x}$

Pullback the adjoint through the graph

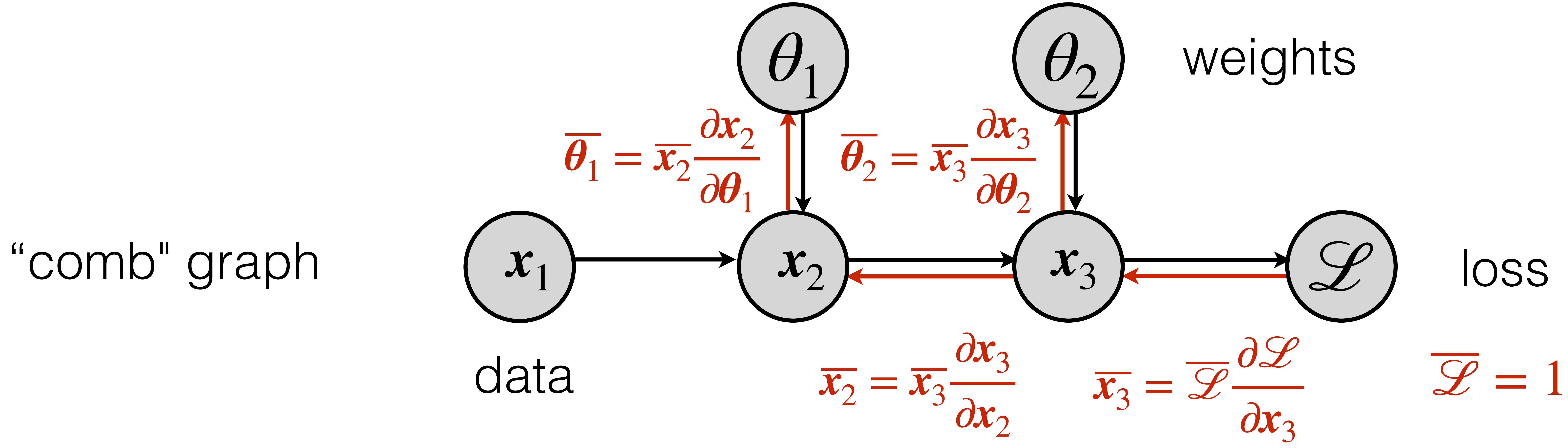
Automatic differentiation on computation graph



Define "adjoint" $\bar{x} = \frac{\partial \mathscr{L}}{\partial x}$

Pullback the adjoint through the graph

Automatic differentiation on computation graph

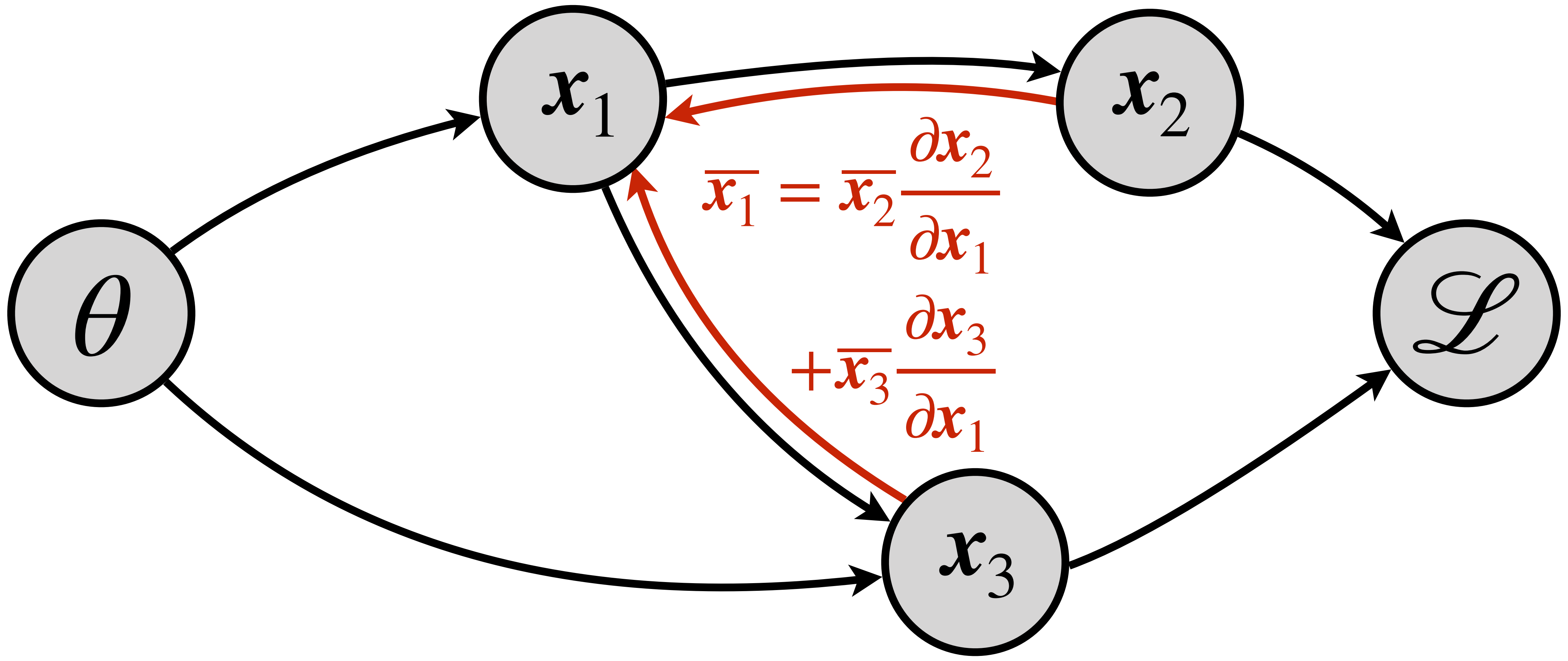


Define "adjoint" $\bar{x} = \frac{\partial \mathscr{L}}{\partial x}$

Pullback the adjoint through the graph

Automatic differentiation on computation graph

directed acyclic graph

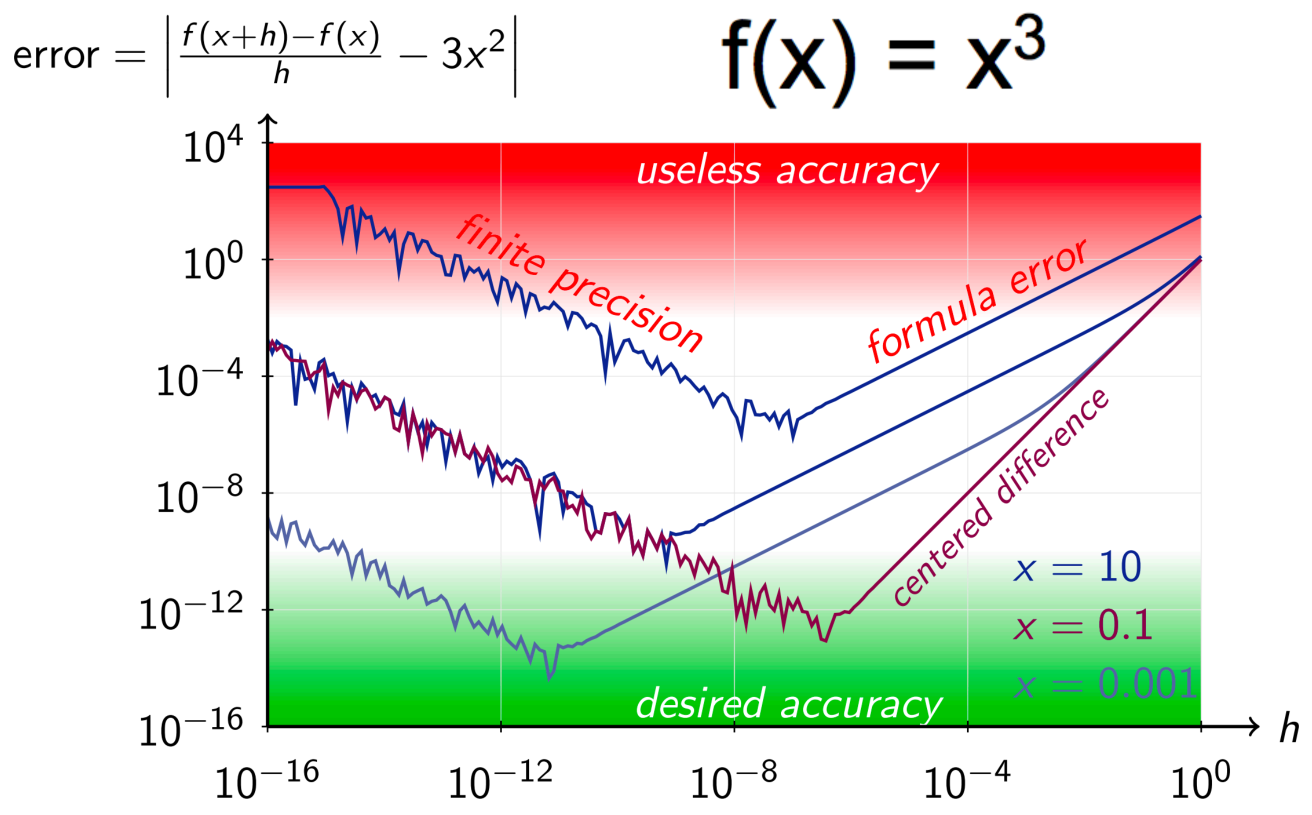


$$\overline{\boldsymbol{x}_i} = \sum_{j:\,\text{child of } i} \overline{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_i} \quad \text{with} \quad \overline{\mathscr{L}} = 1$$

Message passing for the adjoint at each node

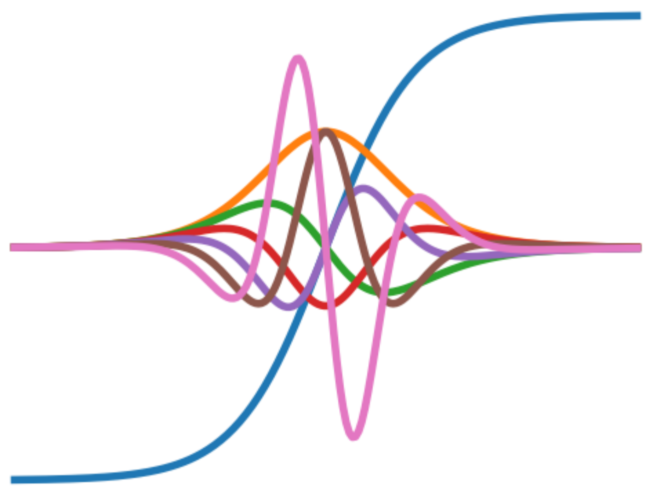
Advantages of automatic differentiation

- Accurate to the machine precision



- Same computational complexity as the function evaluation: Baur-Strassen theorem ’83

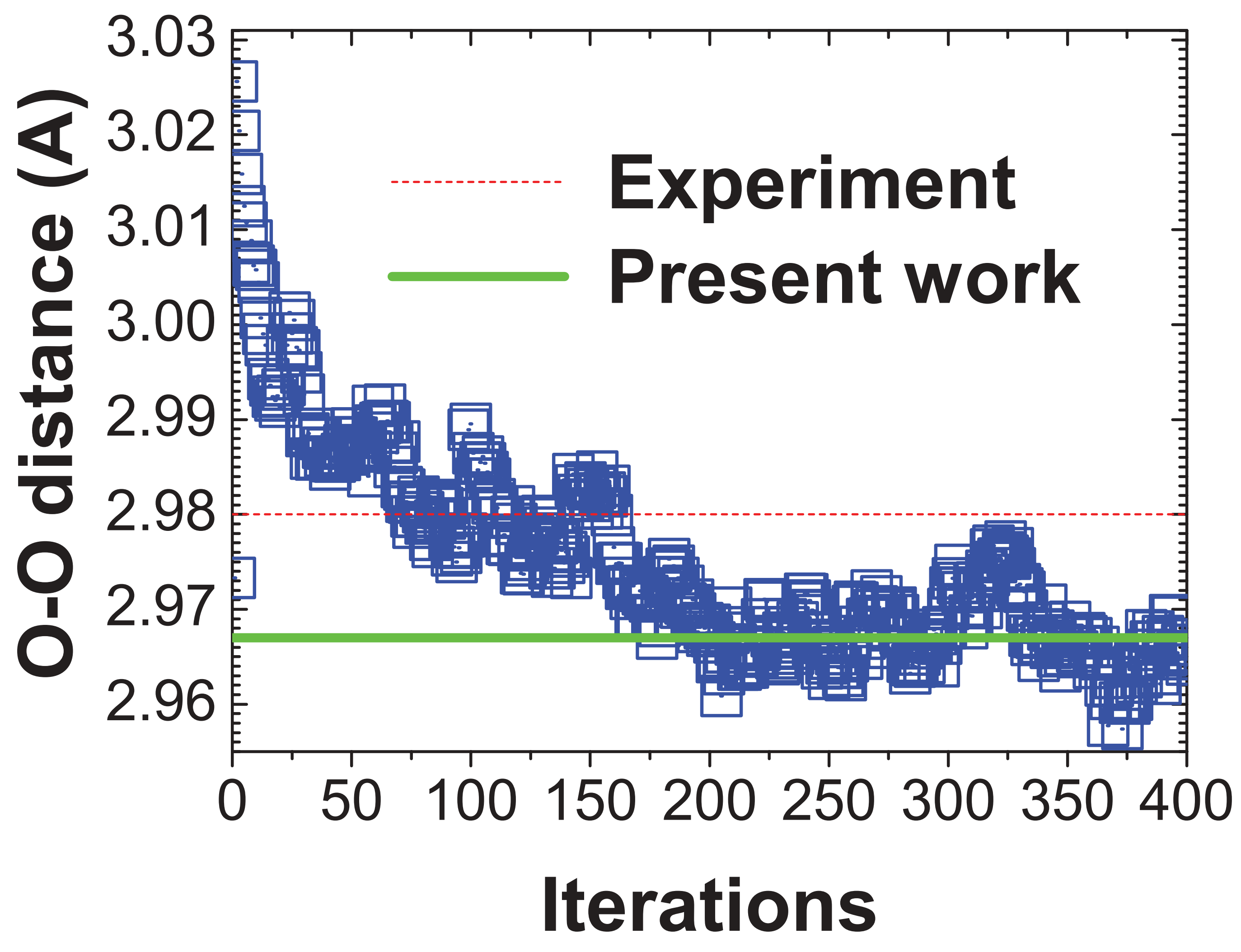
- Supports higher order gradients



```
>>> from autograd import elementwise_grad as egrad  # for functions that vectorize over inputs
>>> import matplotlib.pyplot as plt
>>> x = np.linspace(-7, 7, 200)
>>> plt.plot(x, tanh(x),
...          x, egrad(tanh)(x),                                     # first  derivative
...          x, egrad(egrad(tanh))(x),                              # second derivative
...          x, egrad(egrad(egrad(tanh)))(x),                       # third  derivative
...          x, egrad(egrad(egrad(egrad(tanh))))(x),                # fourth derivative
...          x, egrad(egrad(egrad(egrad(egrad(tanh)))))(x),         # fifth  derivative
...          x, egrad(egrad(egrad(egrad(egrad(egrad(tanh))))))(x))  # sixth  derivative
>>> plt.show()
```

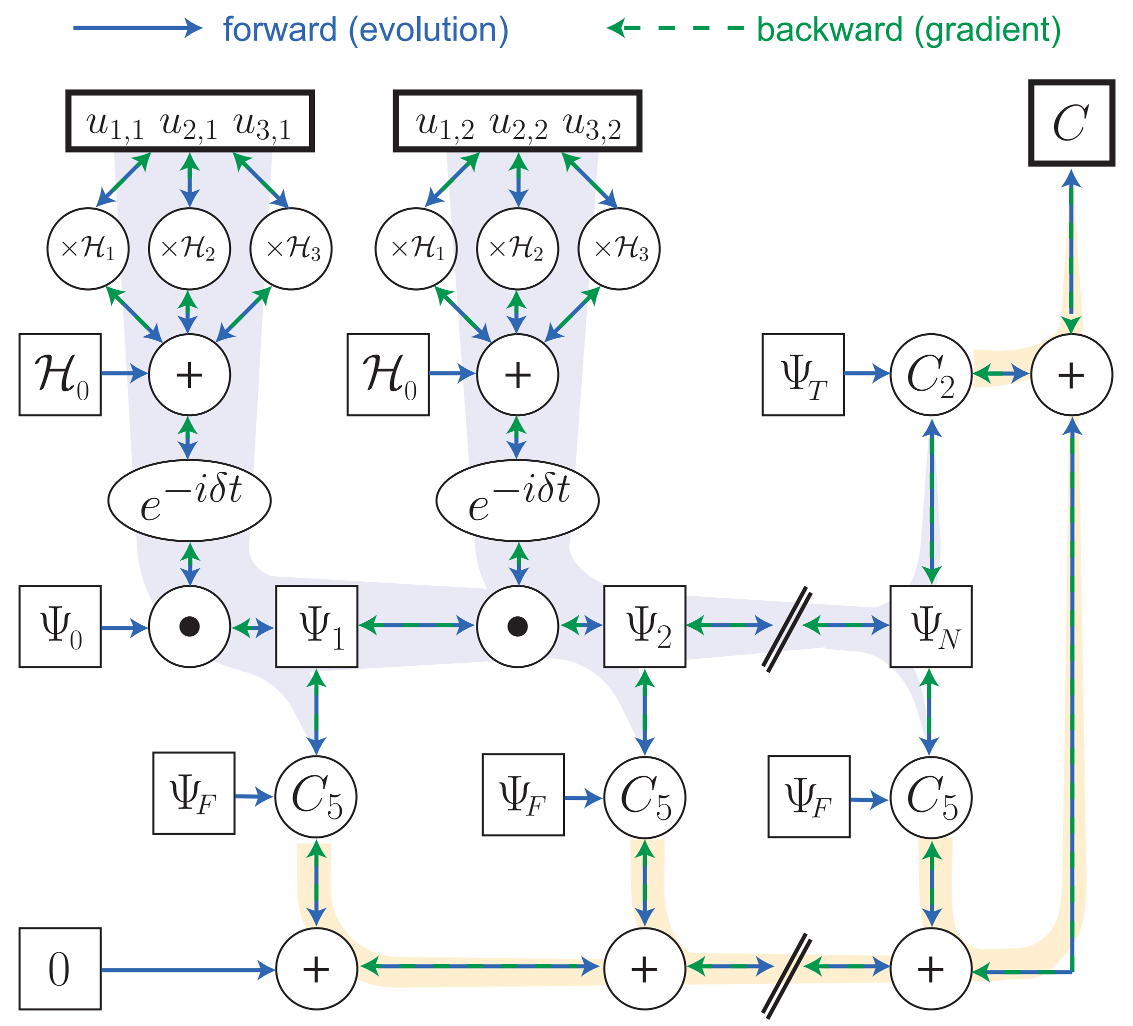
Applications of AD

Computing force



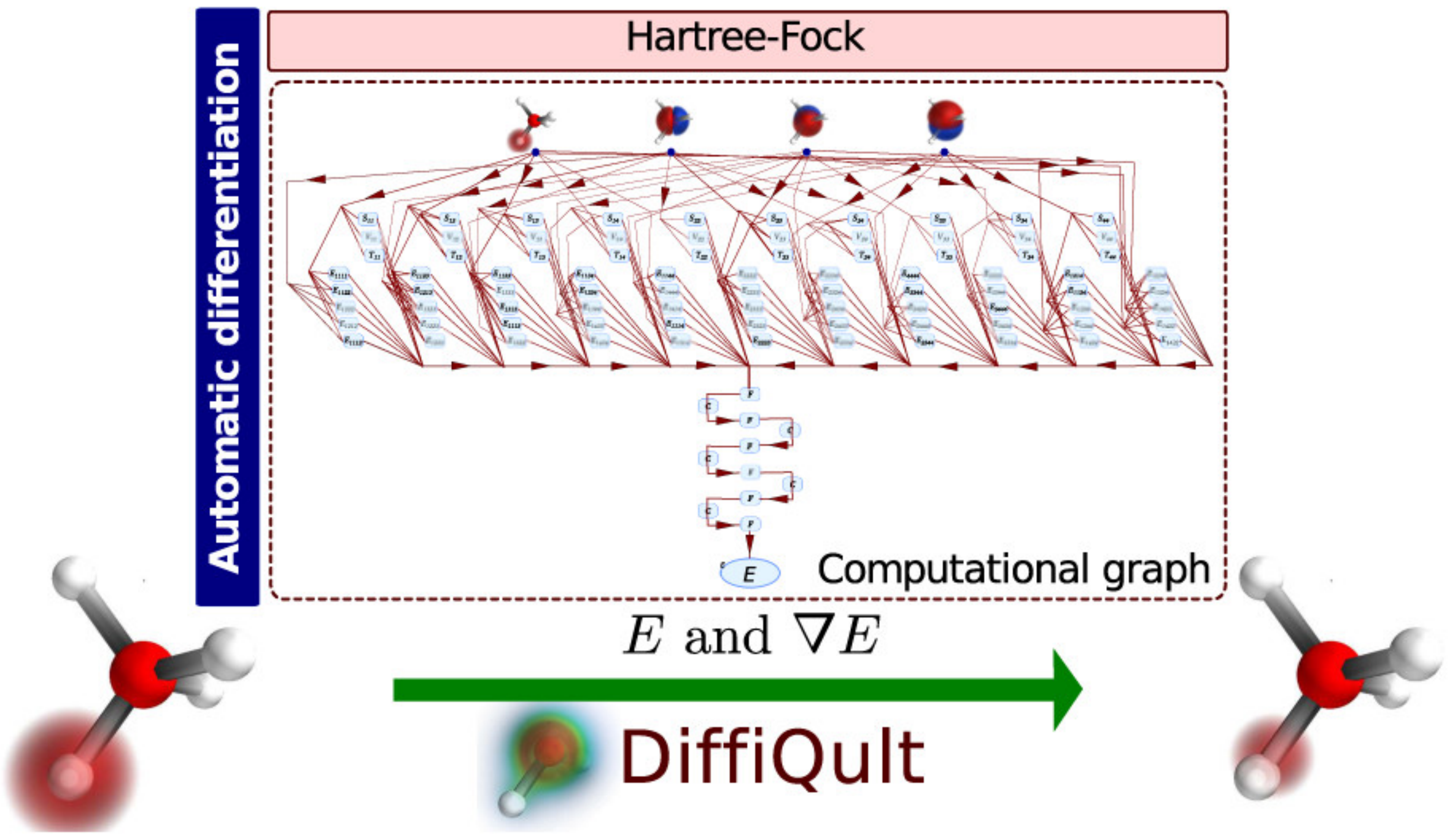
Sorella and Capriotti
J. Chem. Phys. ’10

Quantum optimal control



Leung et al
PRA ’17

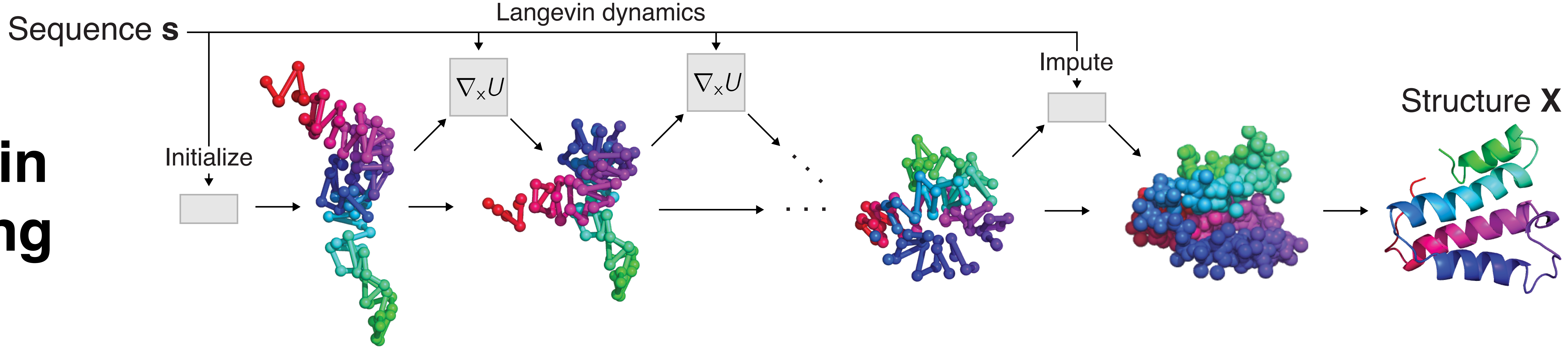
Variational Hartree-Fock



Tamayo-Mendoza et al
ACS Cent. Sci. ’18

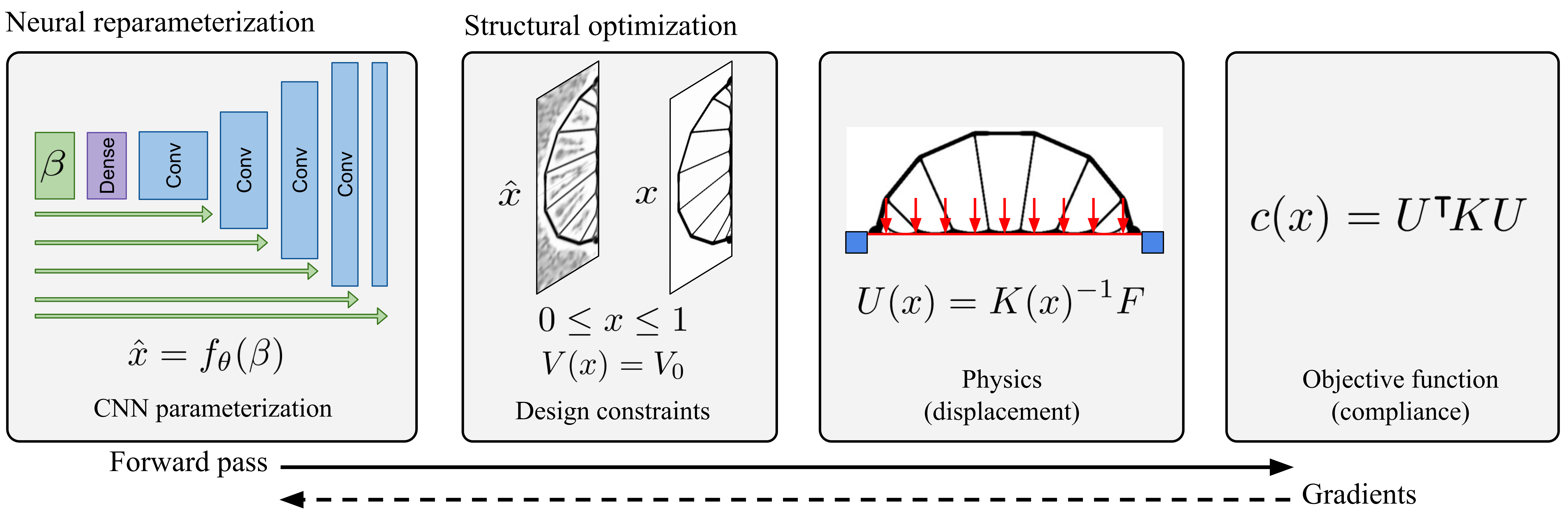
More Applications…

Protein Folding



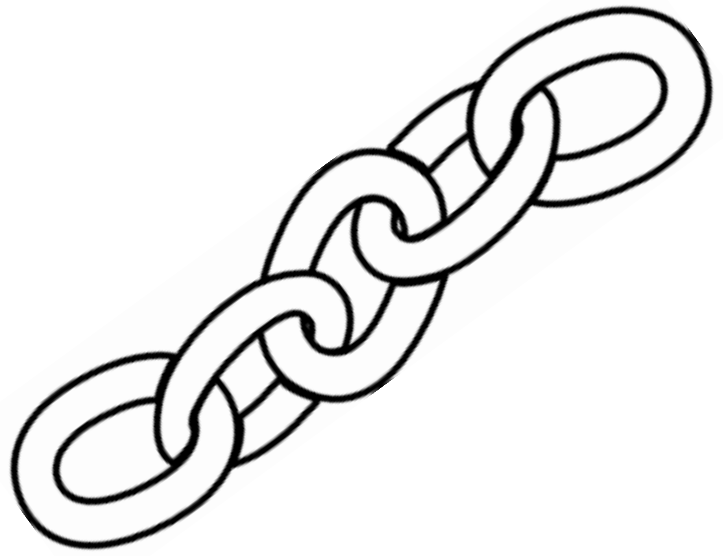
Ingraham et al ICLR '19

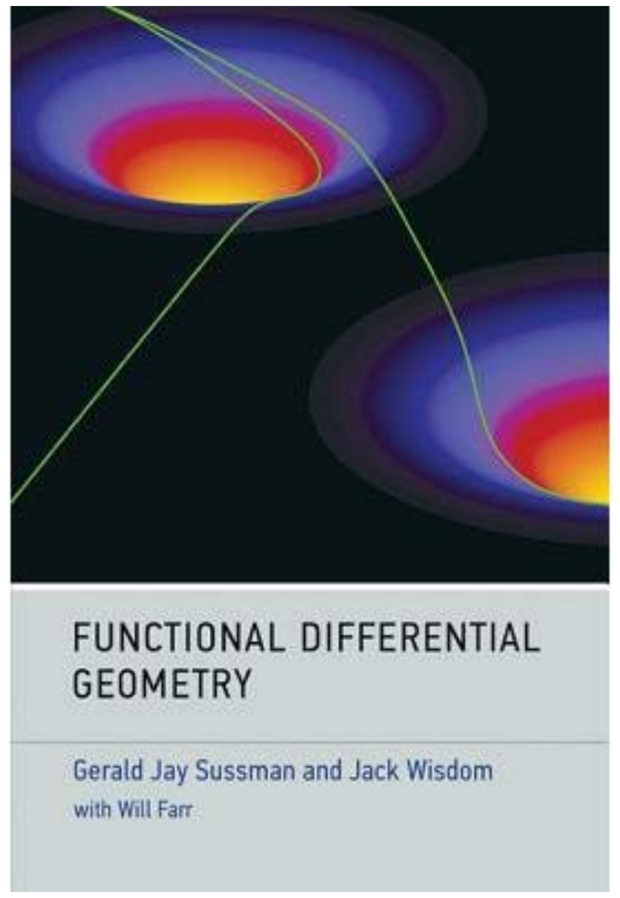
Structural Optimization



Hoyer et al 1909.04240

Understandings of AD





Black
magic box

Chain
rule

Functional
differential geometry

https://colab.research.google.com/github/google/jax/blob/master/notebooks/autodiff_cookbook.ipynb

Reverse versus forward mode

$$\frac{\partial \mathscr{L}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathscr{L}}{\partial \boldsymbol{x}_n} \frac{\partial \boldsymbol{x}_n}{\partial \boldsymbol{x}_{n-1}} \cdots \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1} \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{\theta}}$$

$$\longrightarrow$$

Reverse mode AD: Vector-Jacobian Product of primitives

- Backtrace the computation graph $\quad v_o (J)_{o \times i}$
- Needs to store intermediate results
- Efficient for graphs with large fan-in

Backpropagation = Reverse mode AD applied to neural networks

Reverse versus forward mode

$$\frac{\partial \mathscr{L}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathscr{L}}{\partial \boldsymbol{x}_n} \frac{\partial \boldsymbol{x}_n}{\partial \boldsymbol{x}_{n-1}} \cdots \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1} \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{\theta}}$$

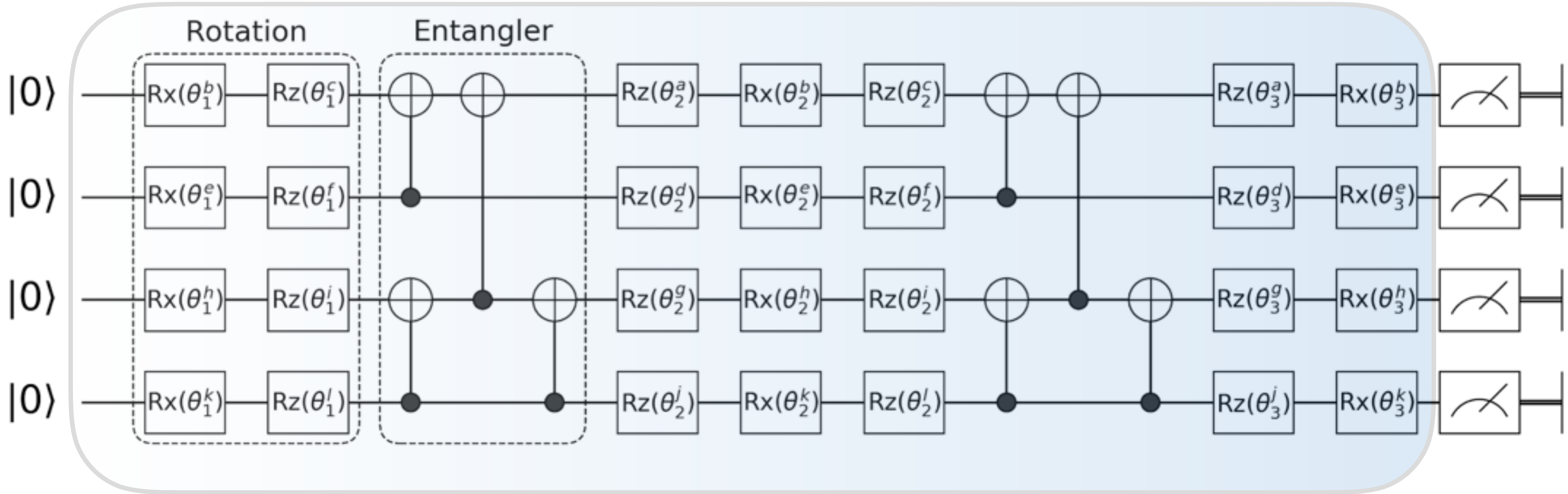
$$\longleftarrow$$

Forward mode AD: Jacobian-Vector Product of primitives

- Same order with the function evaluation $(J)_{o \times i} v_i$
- No storage overhead
- Efficient for graph with large fan-out

Less efficient for scalar output, but useful for higher-order derivatives

Differentiable² quantum circuits



Parametrized gate of the form

$e^{-\frac{i\theta}{2}\Sigma}$ with $\Sigma^2 = 1$

eg, X, Y, Z, CNOT, SWAP…

Li et al, PRL '17, Mitarai et al, PRA '18
Schuld et al, PRA '19, Nakanishi et al '19

$$\nabla \langle H \rangle_\theta = \left(\langle H \rangle_{\theta+\pi/2} - \langle H \rangle_{\theta-\pi/2} \right) / 2$$

Unbiased gradient estimator measured on actual quantum circuits

Demo 3

https://github.com/wangleiphy/YaoTutorial

Applications of Yao.jl

Quantum machine learning:

Differentiable Learning of Quantum Circuit Born Machine, 1804.04168

Learning and Inference on Generative Adversarial Quantum Circuits, 1808.03425

…

Quantum many-body physics:

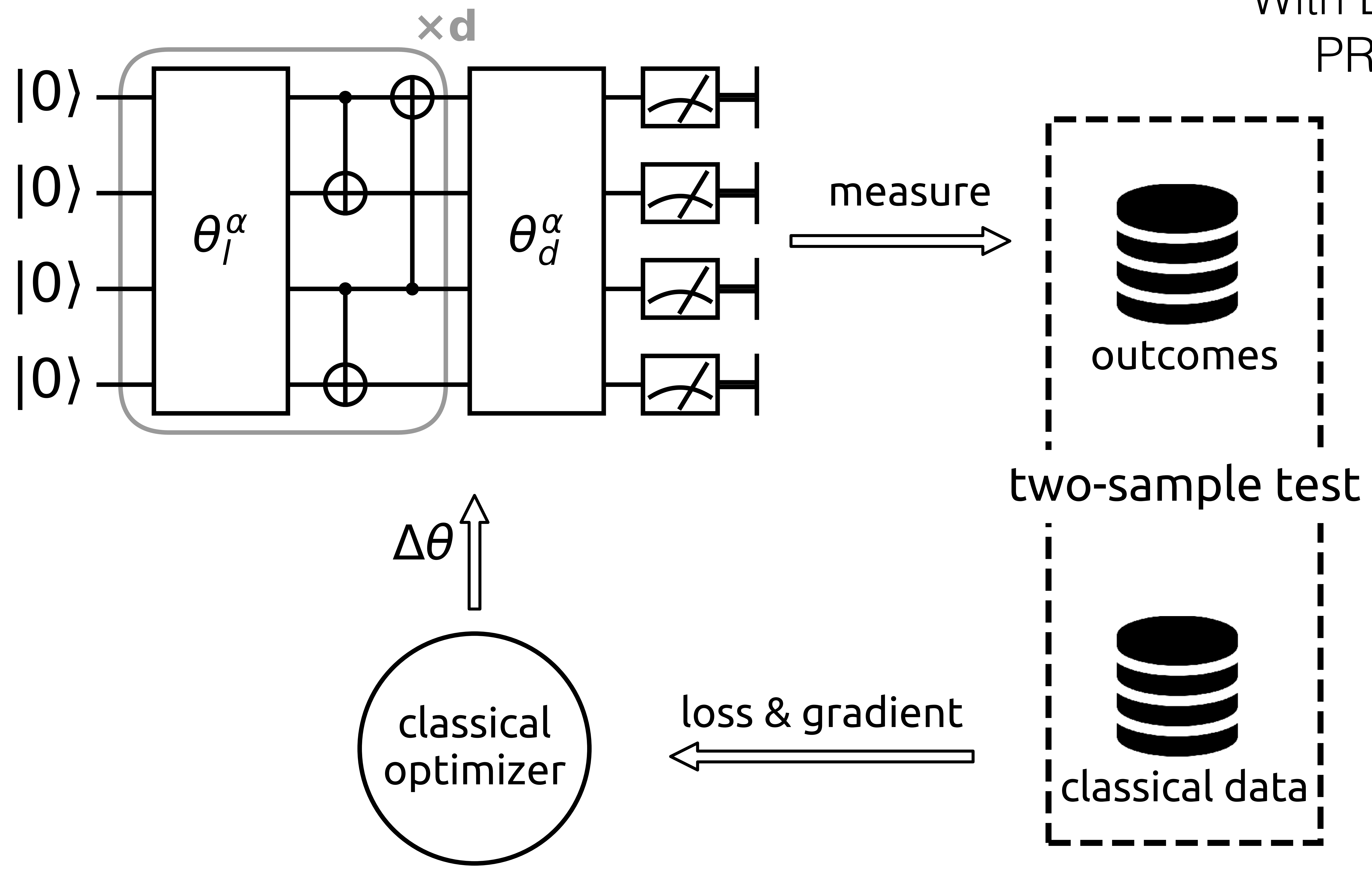
Variational Quantum Eigensolver with Fewer Qubits, 1902.02663

Solving Quantum Statistical Mechanics with VAN + Quantum Circuits, 1912.?????

…

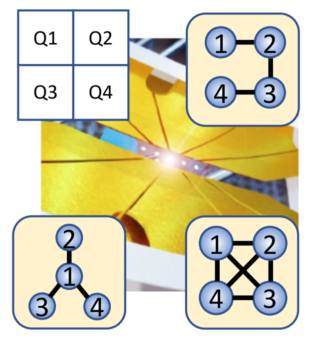
Quantum Circuit Born Machine

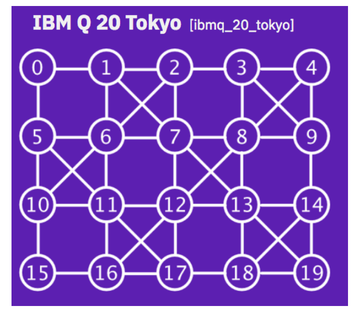
With Liu, Zeng, Wu, Hu
PRA ’18, PRA ’19

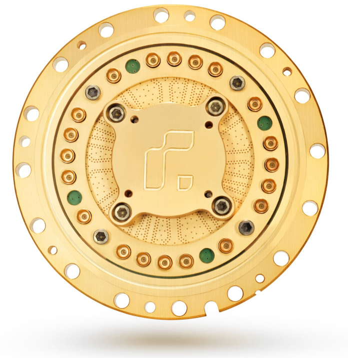


Experiments:

1801.07686
1812.08862
1811.09905
1901.08047
1904.02214







Train quantum circuits as probabilistic generative models with implicit density

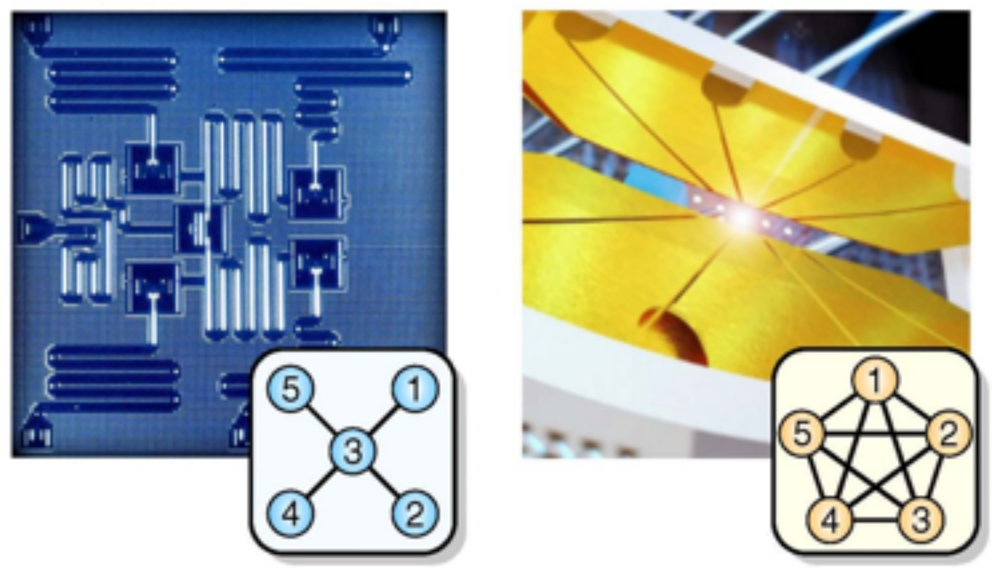
Strong expressibility due to quantum sampling complexity

However, there is a HUGE GAP in the qubit number

What we want to solve

to infinity and beyond

What current technology offers

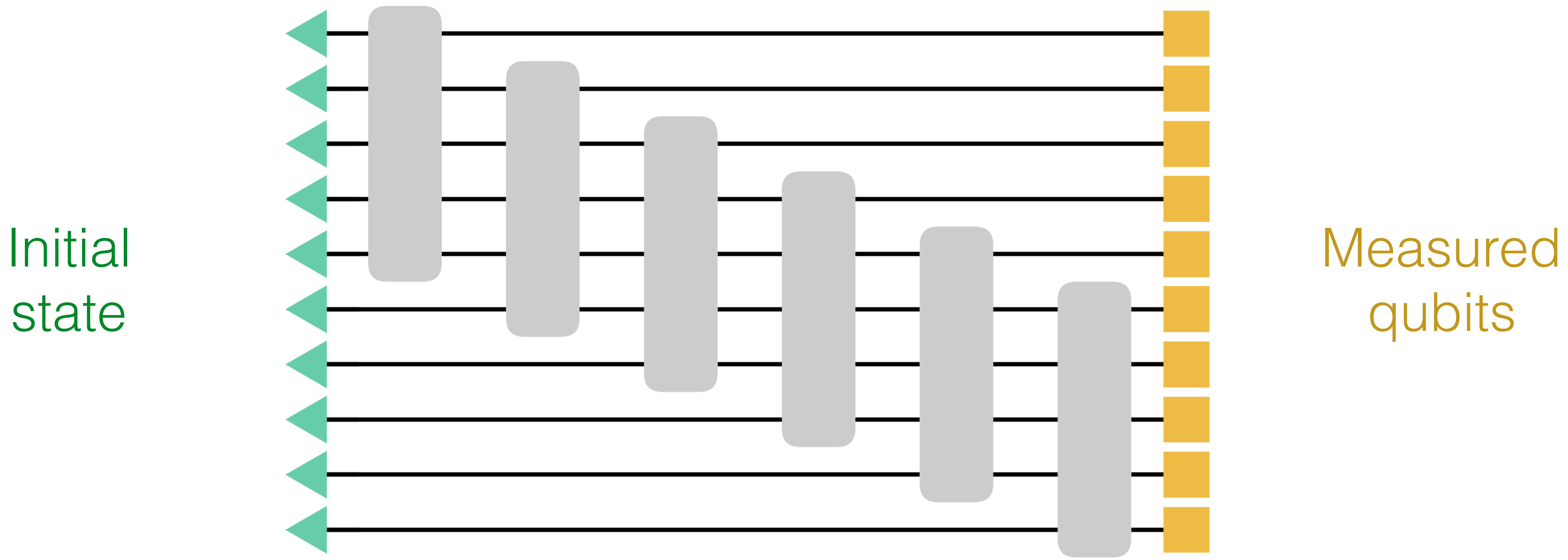


a handful of qubits

Variational quantum eigensolver with fewer qubits

Jin-Guo Liu, Yi-Hong Zhang, Yuan Wan, LW, 1902.02663

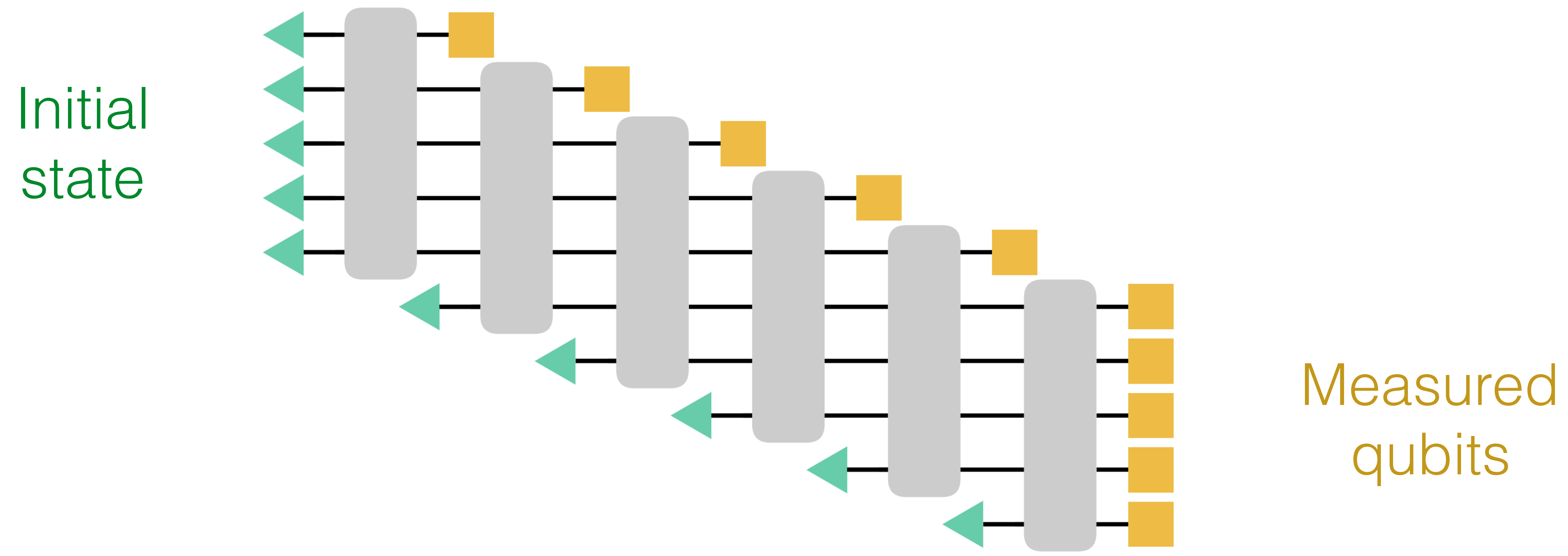
A qubit efficient variational circuit



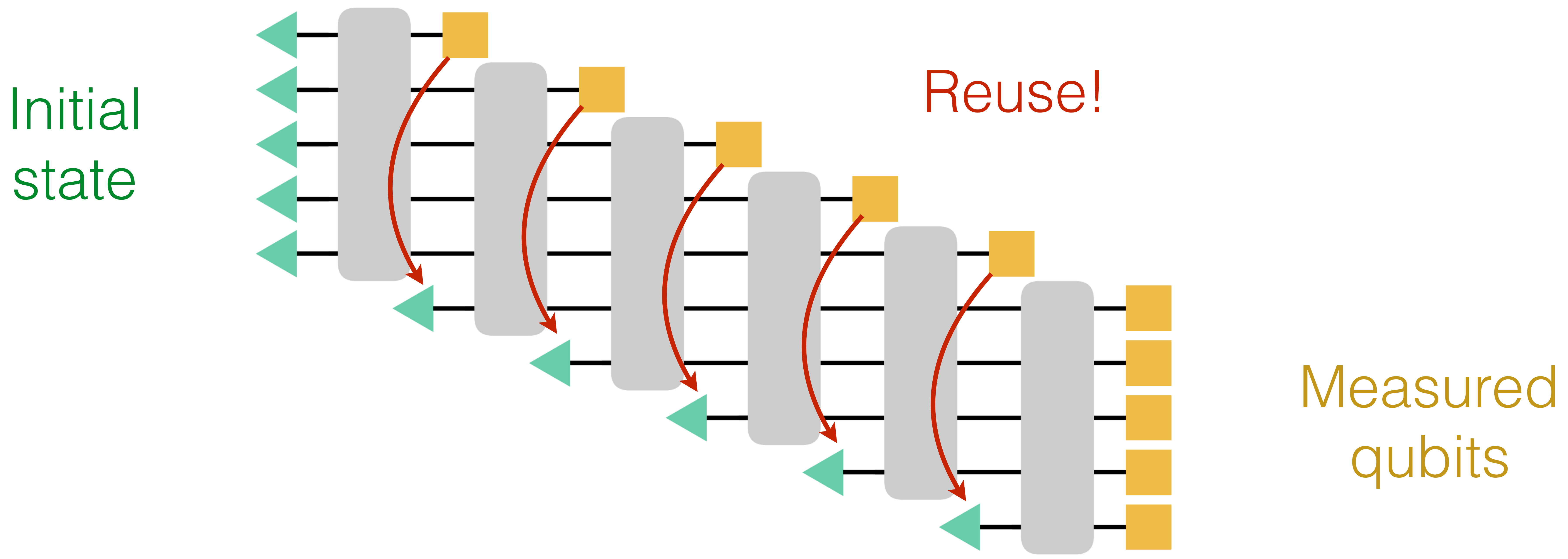
Huggins, Patel, Whaley, Stoudenmire, 1803.11537
see also Cramer et al, Nat. Comm. '10

Tensor network inspired quantum circuit architecture

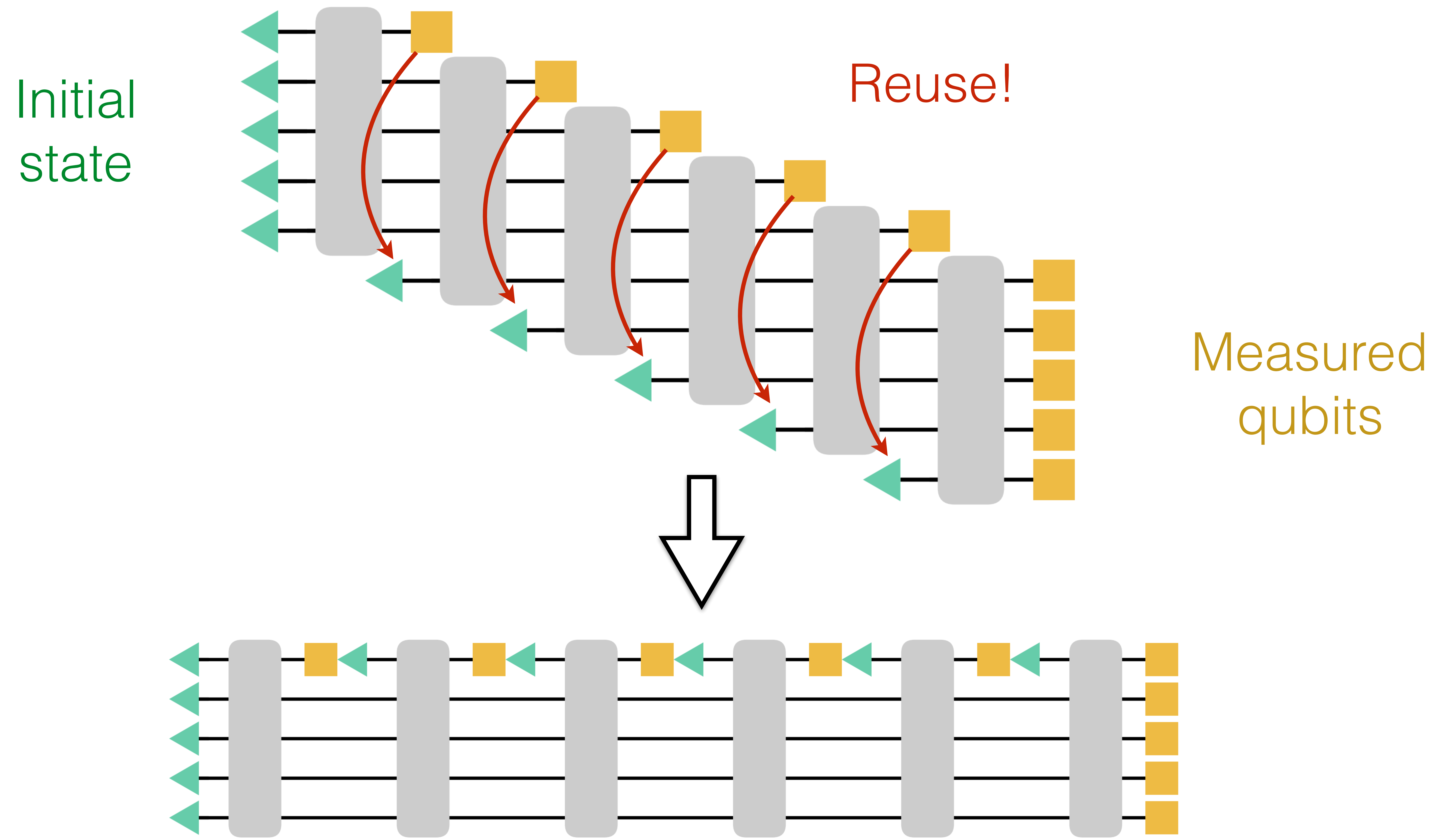
A qubit efficient variational circuit



A qubit efficient variational circuit

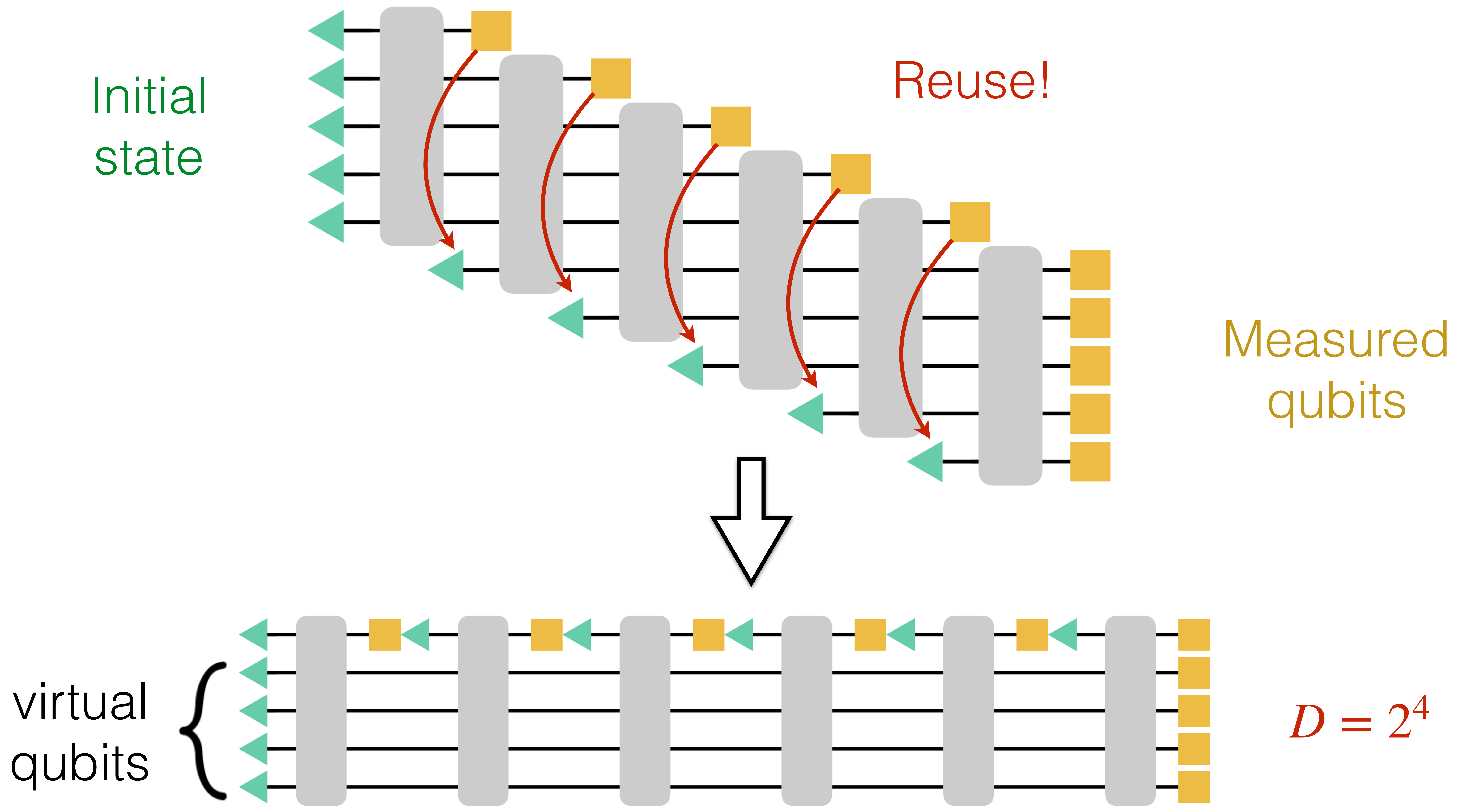


A qubit efficient variational circuit



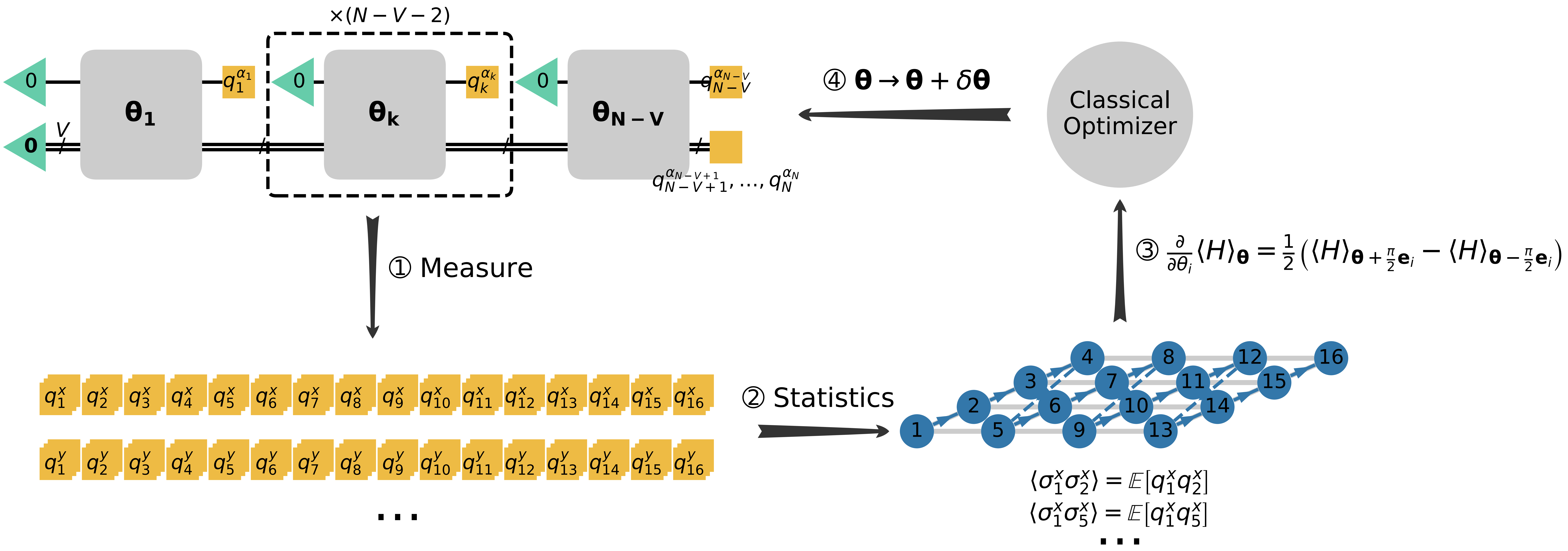
Matrix Product State with exponentially large bond dimensions

A qubit efficient variational circuit



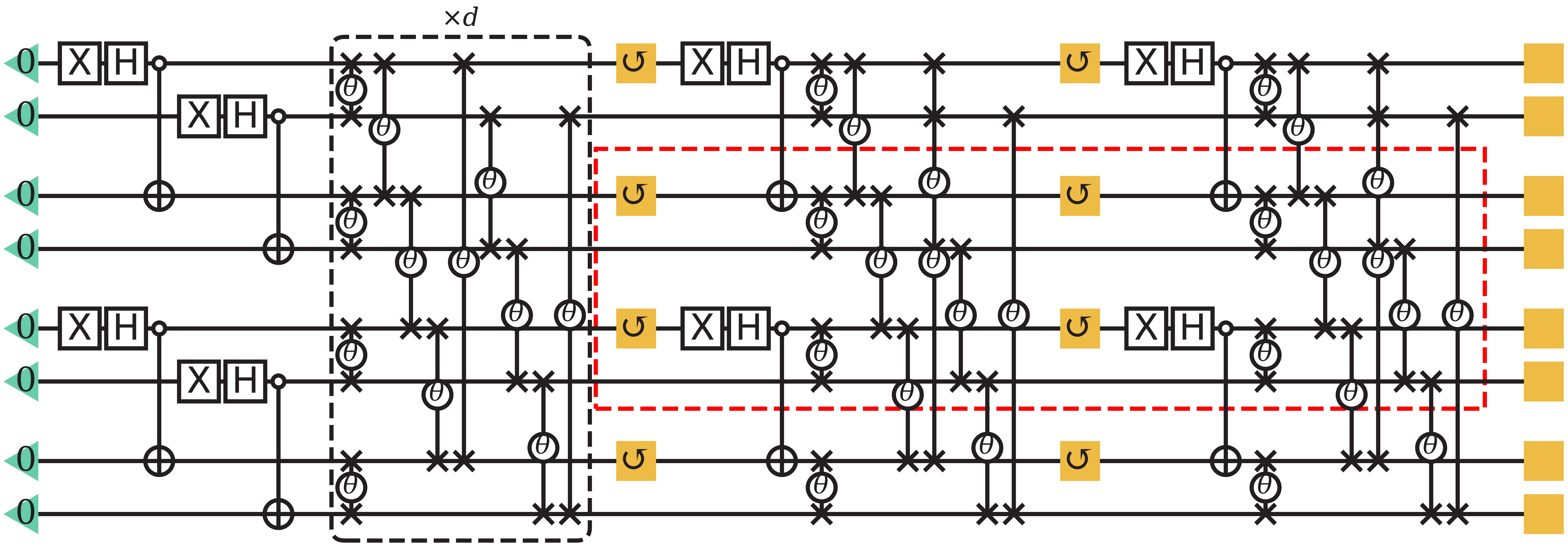
Matrix Product State with exponentially large bond dimensions

Q-MPS



×(N − V − 2)
0
θ1
θk
θN − V
V
④ θ → θ + δθ
Classical Optimizer
① Measure
③ $\frac{\partial}{\partial\theta_i}\langle H\rangle_{\boldsymbol{\theta}} = \frac{1}{2}\left(\langle H\rangle_{\boldsymbol{\theta}+\frac{\pi}{2}\mathbf{e}_i} - \langle H\rangle_{\boldsymbol{\theta}-\frac{\pi}{2}\mathbf{e}_i}\right)$
② Statistics
$\langle\sigma_1^x\sigma_2^x\rangle = \mathbb{E}[q_1^x q_2^x]$
$\langle\sigma_1^x\sigma_5^x\rangle = \mathbb{E}[q_1^x q_5^x]$

Q-PEPS



How to prepare quantum thermal states?

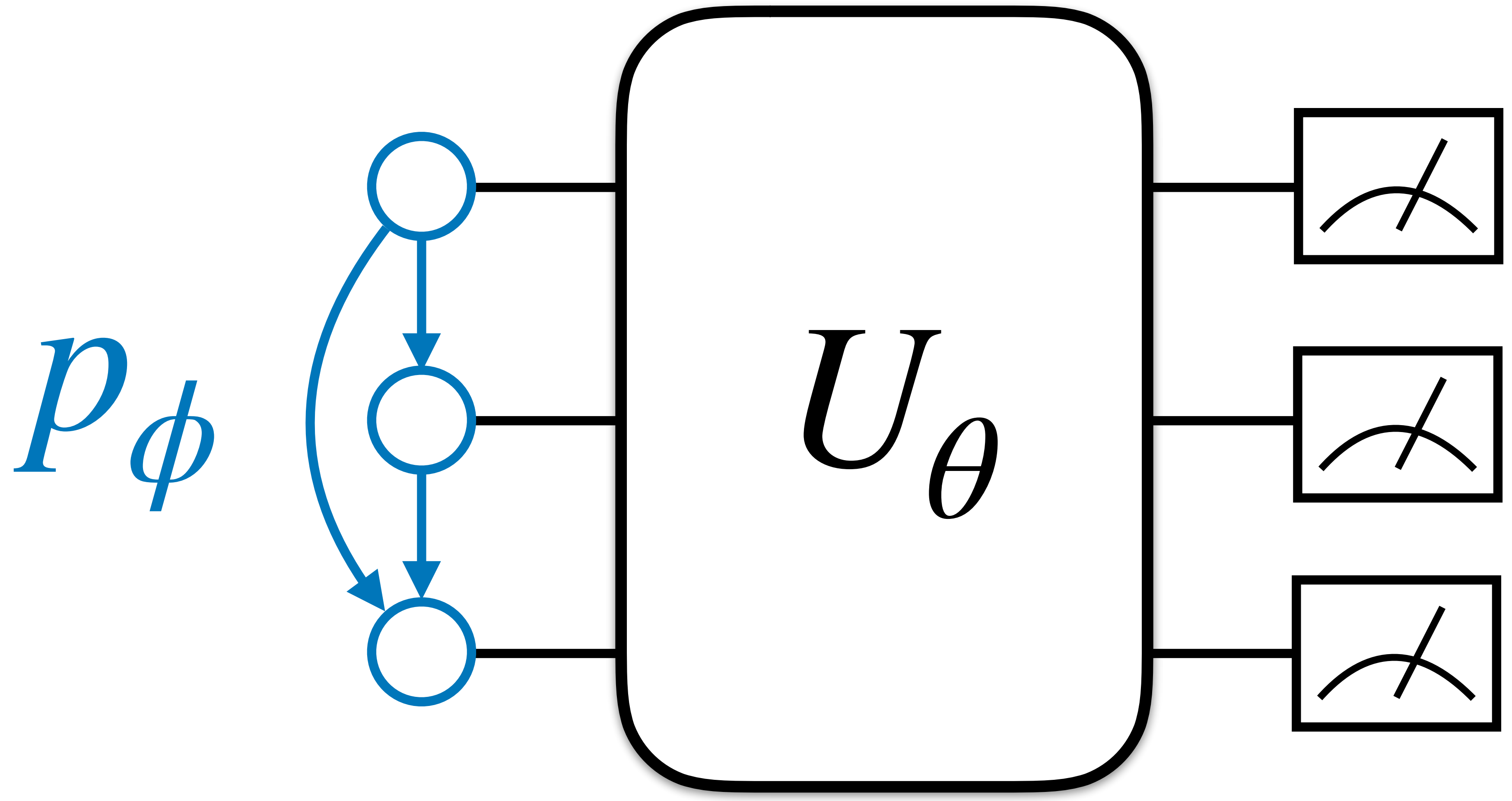
Thermofield Double States

Wu & Hsieh, 1811.11756

Quantum imaginary-time evolution

Motta et al, 1901.07653

"β"-VQE



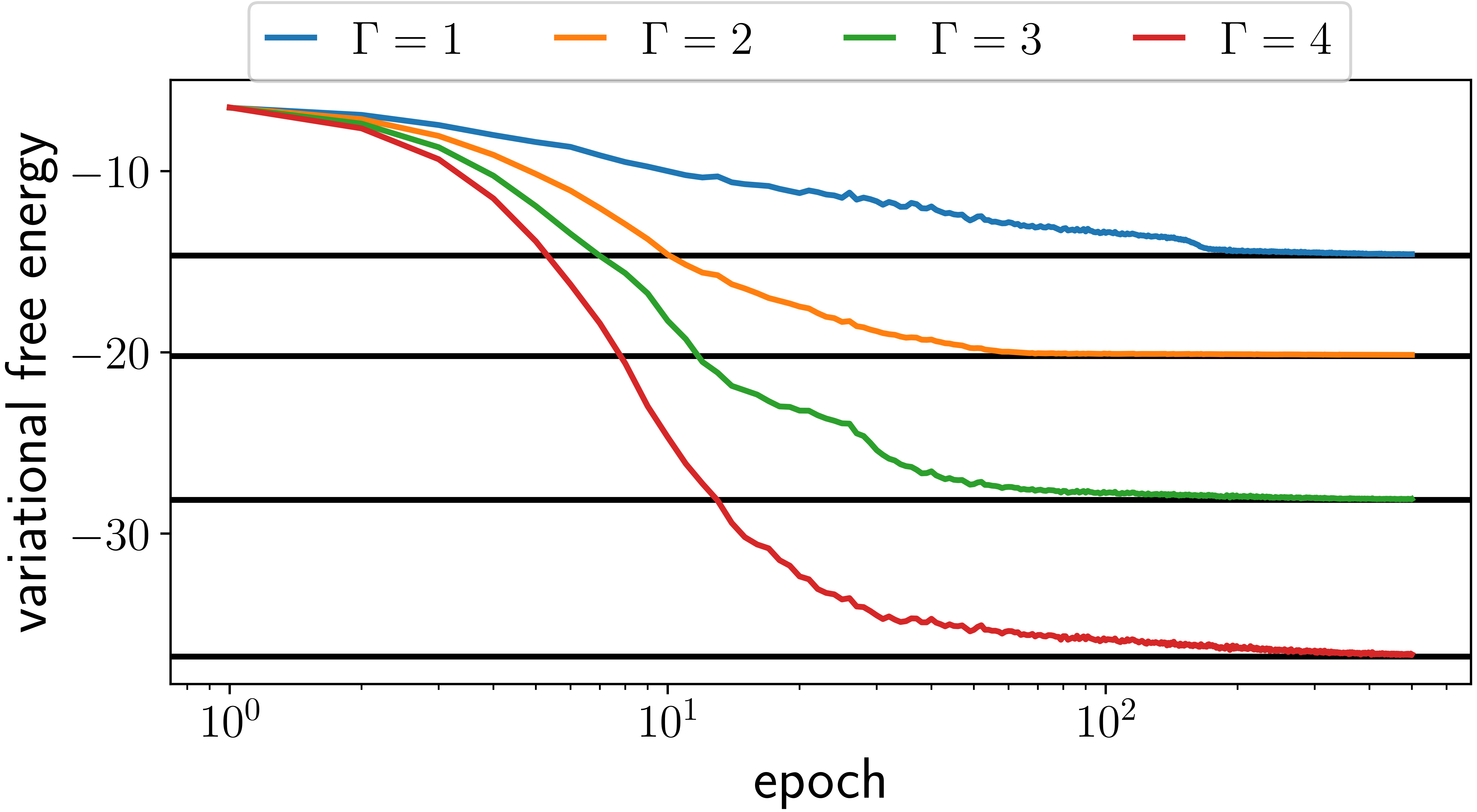
$$\sum_x p_\phi(x) U_\theta |x\rangle\langle x| U_\theta^\dagger = \rho$$

A classical mixture of quantum states parametrizes density matrices

Martyn & Swingle, 1812.01015 Verdon et al, 1910.02071

"β"-VQE

$$\mathscr{L} = \beta \mathrm{Tr}(\rho H) + \mathrm{Tr}(\rho \ln \rho) \geq -\ln Z$$



3x3 quantum
Ising model @ β=1

Liu, Mao, Zhang,
LW, 1912.11381

Study quantum thermodynamics with classical & quantum flows

```
julia> using Yao
```

Yao offers you freedom no one else can offer

Make your own innovation in quantum algorithms design!

Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design

Xiu-Zhe Luo, Jin-Guo Liu, Pan Zhang, Lei Wang, 1912.10877

Thank You!